

**This Page Is Inserted by IFW Operations
and is not a part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

~~Best Available Copy~~

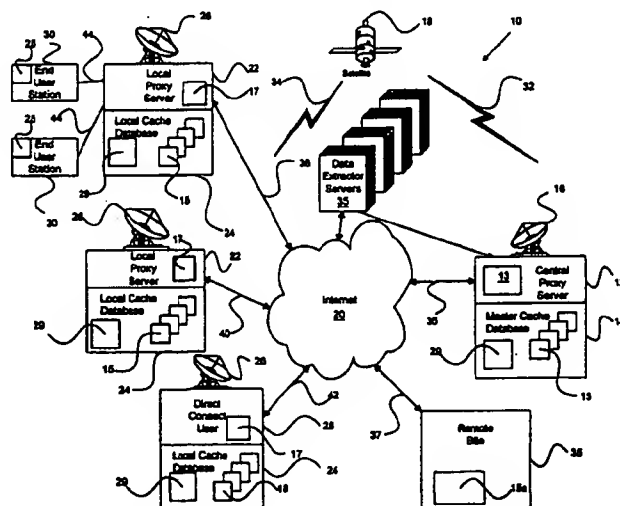
THIS PAGE BLANK (USPTO)



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 15/16		A1	(11) International Publication Number: WO 00/42519
			(43) International Publication Date: 20 July 2000 (20.07.00)
(21) International Application Number: PCT/US00/00587			(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
(22) International Filing Date: 11 January 2000 (11.01.00)			
(30) Priority Data:			
60/115,456	11 January 1999 (11.01.99)	US	
60/169,971	8 December 1999 (08.12.99)	US	
Not furnished	10 January 2000 (10.01.00)	US	
(71) Applicant: EDGIX CORPORATION [US/US]; 1322 East 550 North, Orem, UT 84097 (US).			Published <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>
(72) Inventors: POWELL, Kyle; 1322 East 550 North, Orem, UT 84097 (US). HURST, Mark; 6222 West 10480 North, Highland, UT 84003 (US). MERKLEY, Karl; 1104 East 230 North, Lindon, UT 84042 (US).			
(74) Agent: WITT, Evan; Madson & Metcalf, 15 West South Temple, Suite 900, Salt Lake City, Ut 84101 (US).			

(54) Title: INTERNET CONTENT DELIVERY ACCELERATION SYSTEM



(57) Abstract

A system and method accelerates distribution of content of a global communications network such as the Internet (20). A central proxy server (12) transmits extracted data files over a communications medium to provide content filling of local proxy servers (17). The communications medium may be satellite transmission and may be conducted using IP multicast. The local proxy servers concurrently receive the data files from the communication medium at a high rate of speed and store the data files in attendant local cache databases (29). The local proxy servers utilize a localized heuristics scheme to determine whether to keep or discard the data. When a user requests Internet data, the user's request is first received by the local proxy server. If the requested data is present among the extracted data files, it is rapidly transmitted directly to the user from the local proxy server, if not present, a request is transmitted to the central proxy server.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

INTERNET CONTENT DELIVERY ACCELERATION SYSTEM

BACKGROUND OF THE INVENTION

1. The Field of the Invention

5 The present invention relates to Internet acceleration systems. More specifically, the present invention relates to Internet Acceleration systems involving local and remote caching to speed up distribution of content on the Internet.

2. The Relevant Art

10 Gridlock has hit the Internet--too much Internet traffic and too little bandwidth are resulting in slow response times. One solution being developed in response to this problem is caching. It is estimated that 60 to 80 percent of Internet traffic is cachable. Taking the web pages hit most often, and locating them near the edge of the network and close to users at the business or ISP site dramatically reduces Internet traffic. With caching solutions, storage is
15 used to improve Internet performance instead of adding communication bandwidth. This has important implications because the cost of storage continues to fall while the cost of bandwidth remains relatively expensive.

 While caching solutions can significantly boost Internet performance, two major problems remain. First, while backbone operators that sell directly to large customers have
20 sufficient "critical mass" to benefit significantly from caching, businesses and smaller ISPs do not. The success of web page caches is a function of "hit rate" (percentage of requests where data is already in cache). But there is an enormous amount of web pages available and in a small population of varied end users there is a much smaller probability that the request of a particular end user will already be cached than with large populations of end users. The
25 second problem is that in order to fill the cache, web pages still transit the Internet backbone, just not as often. With current caching systems, only subsequent references to web pages avoid the Internet connection and have faster access times.

 Solutions for improving caching systems are needed. Among those solutions, improved systems for increasing hit rates within caches would be a great improvement in the
30 art. Additionally, improved manners of extracting web data and distributing that data to local caches would be very helpful. Particularly helpful would be manners of communicating with local caches to determine which web pages are most popular and transmitting that popularity information to a central location where it can be compiled and used to select web data for

transmission to local caches. Additionally, the inventors believe it would be helpful to be able to communicate closely with local caching systems. This would be further beneficial in that it would provide the ability to extract diverse types of content popularity and characteristics for use in constructing global popularity data.

5

OBJECTS AND BRIEF SUMMARY OF THE INVENTION

The Internet content delivery acceleration system of the present invention has been developed in response to the present state of the art, and in particular, in response to the problems and needs in the art that have not yet been fully solved by currently available Internet content delivery acceleration systems. Accordingly, it is an overall object of the present invention to provide a system that overcomes many or all of the above-discussed shortcomings in the art.

To achieve the foregoing object, and in accordance with the invention as embodied and broadly described herein in the preferred embodiment, an improved Internet content delivery acceleration system is provided. Within the system, a central proxy server, or caching system, gathers high demand Internet data and disseminates that data to local proxy servers. The dissemination of data may be broadcast, multicast, reliable multicast, unicast, and reliable point to point transport on any suitable medium over any suitable medium, including over the electromagnetic waves, fiber optics itself and over Satellite. In one embodiment, the transmission comprises multicast distribution, such as IP multicast. Thus, local content filling of the local proxy servers is provided by transmitting Internet data to all subscribing local proxy servers at a high rate of speed. Through the use of multicast protocol, only subscribing or interested local proxy servers receive the transmission.

After the passage of a selected amount of time, the local proxy servers utilize a locally customizable heuristics determination to determine whether to keep or discard the data. The heuristics determination preferably employs global popularity data, and may also comprise customized local standards or information relevant to the particular customers subscribing to the local proxy server.

The local proxy servers are preferably provided with software that enables them to receive the data at the high rate of speed and to store the data in attendant local cache databases for quickly servicing future Internet data requests. In one embodiment, the software comprises a cache database management module configured to communicate directly with a caching database. The local proxy server software preferably also comprises a local caching

module in integral communication with the cache database management module. Preferably, the integral communication comprises direct communication between programs on a common server or other computer, and may be facilitated by an applications program interface (API).

5 Because of the tight integration between the cache database management module and the local caching module, sophisticated heuristics determinations may be employed. For instance, rather than simply registering and reporting to the central proxy server when a data file is requested and is not present in a local cache, customized hit information for all transmitted data files and even data files that have not been transmitted may also received from the caching databases and transmitted to the central proxy server for analysis and use in
10 determining which data files to obtain and forward to the local proxy servers. Additional determinations may be made regarding characteristics such as the timing of demand for data files, together with the nature of the data file for customized determinations of demand within the individual local proxy servers.

When a user requests Internet data, the request is first sent to the local cache database
15 management module to determine whether the requested data is present. If the data is present, it is rapidly transmitted to the user from the local cache database, preferably avoiding any transmission over the Internet. If the data is not present, a request is transmitted to the central proxy server which requests a copy from the hosting site.

The system may also transmit selected data to subscribing local intranet sites to rapidly
20 and systematically publish private data to local proxy servers connected to the local intranets.

BRIEF DESCRIPTION OF THE DRAWINGS

In order that the manner in which the advantages and objects of the invention are obtained will be readily understood, a more particular description of the invention briefly
25 described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

30 Figure 1 is a schematic block diagram illustrating one embodiment of an Internet content delivery acceleration system of the present invention, including a central proxy server and a plurality of remote proxy servers.

Figure 2 is a schematic block diagram illustrating a further embodiment of an Internet content delivery acceleration system of the present invention, including private intranet facilities.

5 Figure 3 is a schematic block diagram illustrating functional components of one embodiment of software used with the systems of Figures 1 and 2.

Figure 4 is a schematic block diagram illustrating a packet for satellite pointcast to local private intranets.

Figure 5 is a schematic block diagram illustrating the functional components of one embodiment of a heuristics procedure of the present invention.

10 Figure 6 is a schematic flow chart diagram illustrating one manner of operation of one embodiment of software for use on the remote proxy server of Figure 1.

Figure 7 is a schematic flow chart diagram illustrating one manner of operation of one embodiment of software for use with the central proxy server of Figure 1.

15 Figure 8 is a schematic block diagram illustrating one embodiment of a central proxy server of the present invention.

Figure 9 is an illustration of the structure of one embodiment of a communications packet suitable for use in conjunction with the present invention.

Figure 10 is a schematic block diagram of one embodiment of a local proxy server of the present invention.

20

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

With reference to Figure 1, shown therein is an Internet content delivery acceleration system 10 of the present invention. Within the system 10 is a central proxy server 12. In one embodiment, the central proxy server 12 comprises a network server 11, such as a
25 personal computer (PC) operating under a network protocol such as the IPX protocol. Shown operating within the central proxy server 12 is an attendant master cache database 14. The master cache database 14 stores frequently used data files 15 extracted from the Internet by the central proxy server 12. The data files 15 in the embodiment of Figure 1 comprises Internet web pages 15a. The data files 15 may comprise any form of Internet
30 content, including HTML files, video and music files, and any other digitally represented information that is transmitted across the Internet.

A central caching module 13 preferably operates within the central proxy server 12. Also included in the depicted embodiment of the system 10 is a plurality of local proxy

5 servers 22, each having a local caching module 17 disposed therein. Each local proxy server 22 also comprises or is otherwise in communication with a local cache database 24. The local proxy servers 22 in one embodiment are high-performance, high-capacity PC-based servers. In one embodiment, the proxy servers operate under an IPX protocol, such as Novell's Netware™. Facilities in which the local proxy servers 22 may be installed include Internet service providers (ISPs), large and medium businesses, and eventually, as economies of scale make the service highly affordable, small businesses and homes.

10 The local proxy servers 22 are each provided with a local cache database 24. Preferably, the local cache databases 24 are provided with high capacity memory. In one embodiment, the cache databases 24 are provided with hard drive memory in excess of 40 Gigabytes. A caching database management module 29 is provided to interface with the local cache database 24. In one embodiment, the caching database management module comprises an Internet caching system (ICS) program 29, such as Novell Corporation's Border Manager™ and/or ICS™ software. Preferably, the local proxy servers are interfaced
15 in integral communication with the caching database management modules, as will be described in greater detail below.

The local proxy servers 22 preferably provide users at end stations 30 with access to a global communications network, such as the Worldwide Web existing on the Internet 20. This makes them ideally suited for installation at locations such as ISPs, telephone system
20 switching backbones, or points of presence (POPs), and within networks of large companies, such as fortune 500 companies. Thus, an end user at a station 30 may be connected to a local proxy server 22 remotely over a modem, or locally over a network 44. The local proxy servers 22 can be considered to be at the "edge" of the Internet 20, a position allowing the present system 10 to reduce traffic over the conventional transmission
25 routes of the Internet 20.

To do so, the local proxy server servers 22 receive and store high-demand data files 15 in the attendant local cache databases 24. In an embodiment to be described herein, the network data comprises web pages 15a emanating from a remote Internet site 35. The web pages 15a are initially gathered by the central proxy server 12 from Internet sites 35 over a
30 communication channel 36 and are subsequently transmitted from the central proxy server 12 to the local proxy servers 22. The transmission is conducted over a suitable communication medium.

The transmission is referred to herein as a "transmission" over a "communications medium." Nevertheless, this dissemination of information may include the traditional notions of broadcast, as well as multicast, reliable multicast, unicast, and reliable point to point transport on any suitable medium including over electromagnetic waves, electrical cables, fiber optics and satellite.

Under a preferred embodiment of the present invention, the communication medium comprises transmission by a satellite 18 at a high rate of speed enabled by efficient hard drive data receipt and storage methods encompassed within the present invention. In one embodiment, this rate of speed is about 25 Mbps. A preferred manner of transmission is IP multicast.

The web pages 15a are preferably concurrently transmitted to each subscribing local proxy server 22 once any of the local proxy servers 22 requests the web pages 15a. Each local proxy server 22 receives the web pages 15a and stores them in its local cache database 24 for a selected amount of time before considering whether to purge the particular web pages 15a. In one embodiment, the selected amount of time is scaled to the amount of memory residing in the particular local cache database 22.

The local proxy servers 22 in one embodiment utilize proxy caching protocol built into the http Internet language for proxy servers. With this protocol, every time a user at a station 30 requests a web page 15a over the network of which the local proxy server 22 is a part, the request passes through the local proxy server 22. As part of the protocol, the local proxy server 22 initially checks its attendant local cache database 24 to determine whether the web page 15a is present. In one embodiment, this consists of the local caching module 17 checking through the integral communication interface with the cache database management program 29.

If the web page 15a is present, having been transmitted by satellite 18 from the central proxy server 12, the local proxy server 22 immediately transmits the web page 15a through the network 44 to the user at a station 30. Thus, the request is transparent, and to the user at a station 30, it appears as if the web page 15a was transmitted over the Internet 20. Advantageously, by bypassing the Internet 20, the transmission is much more rapid. Additionally, Internet traffic is reduced, and Internet connection costs are potentially much lower.

If the requested web page 15a is not present in the local cache database 24, the http caching protocol causes the request to be passed on through the Internet 20 to the Internet site 35 at which the web page 15a is hosted. The Internet site 35 then transmits the web page 15a over the Internet 20 back to the requesting user at a station 30. This Internet transmission occurs over regular Internet communication channels 38, 40, 42.

Concurrently, the request is also passed to the central proxy server 12. Once the central proxy server 12 receives its copy of the request, it also requests a copy of the requested web page 15a from the hosting Internet site 35. Accordingly, the web page 15a is also transmitted over channels 36, 37 to the central proxy server 12. Once the web page 15a is received, the central proxy server 12 caches the web page 15a within the attendant master cache database 14. It may also transmit the web page 15a to the communicating local proxy servers through the communication medium if the web page 15a is found to have a sufficiently high priority. This transmission in one embodiment is satellite transmission. The web page 15a is transmitted 32 through a satellite uplink transmitter 16 to the satellite 18 at a speed of, for example, 25 Mega bytes per second (Mbps). The satellite 18 then transmits 34 the web page 15a to each of the subscribing local proxy servers 22, at a rate preferably of about 25 Mbps.

The central proxy server 12 may filter the web page 15a before caching it. It may also keep track of the number of requests and transmit web pages 15a (or other data 15) only after a certain minimum number of requests have been logged for that web page 15a. Accordingly, the central proxy server 12 is in a position to be a "Nelson Ratings System" for the Internet 20, having centralized access to web site demand information.

Additionally, the central proxy server 12 preferably receives global popularity information about data files 15 from subscribing local proxy servers 22. This popularity information in a basic embodiment comprises miss data. Additionally, due to the unique configuration of the present invention, more sophisticated information may also be transmitted. This may include, for instance, hit information and timing information, as will be discussed in greater detail below.

The central proxy server may also utilize data 15 popularity information, data 15 characteristics, data associations, and other data useful to local proxy servers 22 in deciding which extracted data 15 to continue to store. Such information may be collected from the

local proxy servers 22, and in addition, may also be received from companion data extractor servers 35 communicating with the central proxy server 12.

5 The data extractor servers 35 preferably locate information relevant to the heuristics determinations of the local proxy servers 22 and pass that information to the central proxy server for compilation and later transmission to the local proxy servers 22 and optionally, to other requesting entities, possibly for a subscription fee. Manners in which this information may be gleaned include directly contacting web sites that contain hit rates or other popularity information. Additionally, the data extractor servers 35 may themselves traverse or "crawl" over the web to examine web sites and observe traffic. When examining web
10 sites, the data extractor servers may follow links on those web sites to other web sites to similarly examine the other web sites. This process may be continuous, and the information that is gathered is preferably passed to the central proxy server, as stated.

Additional information that may be collected by the central proxy server 12 includes the qualitative information regarding the data files 15, rather than just quantitative data.
15 Thus, the types of sites or data 15 may be collected, either from the local proxy servers 22 or from the data extractor servers 35. This data may include, for instance, what the subject matter of the data files 15 is, what types of sites are requesting the data files 15, etc. This information may then be used by the local proxy servers 22 in making a customized heuristics determination according to local demand and local user types.

20 In the depicted embodiment, a single central proxy server services the entire system. Nevertheless, more than one central proxy server 12 may be in operation. For example, different geographical locations may be served by different central proxy servers 12. These multiple central proxy servers 12 may be in communication with each other either through suitable mediums and may share information such as hit and miss rate
25 information and other heuristics-type information.

Each local proxy server 22 is preferably provided with a communication facility, such as a satellite down-link receiver 26. In the depicted embodiment, the down-link receiver 26 receives the satellite multicast 34 of the extracted web pages 15a. Upon receiving the web page 15a, each subscribing local proxy server 22 caches the web page 15a
30 for a selected period of time for access by users at the communicating stations 30. At the end of the selected period of time, each local proxy server 22 preferably utilizes an

individualized local policy and a heuristics program to determine whether to keep the transmitted web page 15a or discard the web page 15a.

Reference is now made to Figure 2 for a discussion of a further aspect of the present invention. In addition to multicasting generally to a series of local proxy servers 22, the
5 central proxy server 12 may also service one or more virtual private intranets 50 through selective pointcast satellite transmission 74. Under this concept, and in an embodiment to be specifically described, the system 10 preferably operates in substantially the same manner as described above. Accordingly, the system 10 is provided with the central proxy server 12 and a series of local proxy servers 22 configured substantially in the manner described.
10 In addition, the system 10 may also be provided with the private intranet 50. The private intranet 50 is shown communicating over the Internet 20 through an intranet proxy server 60. The intranet proxy server 60 may also function as a local proxy server 22, as discussed above.

As depicted, the private intranet 50 connects to end users at remote stations 52
15 through a communication channel 54. The private intranet 50 is connected with the intranet proxy server 60 through a communication channel 68. The Intranet proxy server is in communication with a satellite down-link receiver 62 through a communication channel 66 and with the Internet 20 with a communications channel 64. One or more of the aforementioned channels may maintain security with a firewall 78.

20 The system 10 may also be provided with a private intranet publisher 70 for generating or otherwise providing private data 55 to be communicated to users at stations 52 within the private intranet 50. In the depicted embodiment, the private intranet publisher 70 is a dedicated server computer communicating over the Internet 20 through a secure channel 72. The secure channel 72 is shown provided with a firewall 78.

25 The private intranet publisher 50 generates or collects the private data 55, which may be business, financial, or any other type of data, and transmits the private data 55 in web page form. The private data 55 is passed in a secure manner through the Internet 20 to the central proxy server 12. The central proxy server 12, through a satellite uplink transmitter 16, then uplinks 32 the private data 55 to the satellite 18. The satellite 18 then
30 transmits the private data 55 to intranet proxy servers 60 of the subscribing virtual private intranet using a focused multicast 74. The same satellite 18 may also transmit 34 other data such as web pages 15a to the regular local proxy servers 22.

All private intranet transmission over the Internet 20 is preferably kept within a firewall 78, preferably through hashing or other encryption of the data being transmitted. The multicast transmission 74 could be transmission by a private frequency, or more preferably, through the same frequency and channels as the normal web site multicasts 34, but with a header placed on each transmitted packet identifying the transmission as private and identifying the designee. In such a case, encryption may be employed, and the remote intranets intended to receive the data may be provided with encryption keys. A master key is preferably retained by the central proxy server 12. Local proxy servers 22 for which the data 55 is not intended thus ignore the data 55. The header could also particularly specify whether the data is private data 55 or global data 15.

Thus, the Virtual Private Intranet of the present invention, unlike prior art broadcast intranets, is one-way, and the private data 55 to be transmitted is collected or otherwise designated by the central publisher 70. In one embodiment, the publisher 70 collects private data 55 as a result of requests from remote intranet sites, possibly over the Internet 20.

One example of the use of a virtual private intranet 50 of the present invention is within a car dealership chain. In such a case, the private data 55 is, for example, sales and pricing data. Financial advising companies may also use the system and pointcast 74 data 55 in the form of market data or financial analysis in another example. Banks may make financial transactions with the system 10. Business might transmit sales information, company policies, advertising materials, etc.

It is preferred that the private data 55 is transparent, once pointcast 74 to the users 52. Thus, the data 15 may be viewed using file transfer protocol of the local network, rather than as an Internet URL site. Accordingly, the users 52, though possibly located across the world from each other, may access the private data 55 from the proxy server as if it were a local part of the remote intranet 50. The private data 55 may be transmitted on demand, but it is contemplated that at least a substantial portion of the private data 55 may be transmitted through the discretion of the Internet Publisher 70, or may comprise standard information, periodically updated and transmitted.

Referring now to Figure 3, shown therein is a schematic block diagram illustrating more specifically the various functional components of one embodiment of software used to control the system 10 of the present invention. The software modules contained in the schematic block diagram of Figure 3 are generally implemented as software instructions,

procedures, routines, or other executable software code. Nevertheless, the modules could also be implemented with other types of programmable logic such as programmable logic arrays (PLAs), ASICs, or even logic circuits or other types of hardware.

5 In order to initialize the system 10 (of Figures 1 and 2) and enable communication links, the depicted components of a system initialization block 80 is employed. The modules of the system enablement block 80 may be contained as software code within the central proxy server 12, or may be distributed amongst the different hardware components of the system 10. Initially, the central proxy server 12 is initialized and brought on-line with a central server enablement module 82. This preferably includes enabling Internet
10 communication over communication channel 36 and enabling satellite communication with the satellite uplink transmitter 16.

Subsequently, the local proxy servers 22 are enabled and brought on-line with a local proxy server enablement module 84. Typically, Internet communications are enabled through communication channels 38, 40, and 42. It is of note that the various local proxy servers 22
15 will generally not all be brought on-line at one time. Typically, the system 10 of the present invention will be provided commercially as a service to which customers subscribe. As new customers subscribe, they are provided with the local proxy server 22 and satellite receiver hardware 26 for an initial fee. A monthly subscription fee may be charged thereafter.

Satellite communications are enabled with a satellite initialization module 86. This
20 may include establishing the communications link between the satellite 18 and the satellite uplink transmitter 16 and between the satellite 18 and the satellite receivers 26, 62. It may also include establishing a proper protocol for satellite communications. For the embodiment of Figure 2, satellite communications for the pointcast transmission must be established with one or more Internet proxy servers 60.

25 The intent of the system 10 is to fill the local cache databases 24 of each subscribing local proxy server 22 with the most highly requested data such that the majority of requests for web pages 15a may be serviced directly by the local proxy servers without having to go to the Internet 20. Accordingly, each new central proxy server that is brought on-line must be filled with the most highly requested web pages 15a. This is conducted at with an initial
30 filling module 88 and may be accomplished in a variety of manners. In one contemplated embodiment, the central proxy server 12 transmits "old" data that has been previously transmitted in between transmitting "fresh" data that has only recently been requested. This

transmission may be conducted in bursts to transmit the entire contents of the master cache database systematically, or according to a selected heuristic formula.

5 A user station block 90 of Figure 3 illustrates the basic functional components and operation of a user station 30 of Figure 1. Through an Internet data request module 92 the user at the end station 30 places a request for data 15. In one embodiment, the data 15 is a web page 15a and the data request module comprises a standard web browser 25 which is connected to the Internet through a local proxy server 22. The data 15 is transparently received from the local proxy server 22 through a data reception module 94. The data reception module 94 may comprise the web browser 25 in conjunction with the http proxy caching protocol which searches local cache databases 24 prior to transmitting a request for Internet data over the Internet 20.

10 Once the web page 15a is received, either from the local cache database 24 or over the Internet 20, it is displayed to the user with a data display module 96. The data display module 96 may comprise the web browser 25 together with the proxy caching protocol of the http language operating on a PC or other computer.

15 A central server block 100 illustrates the basic functional components operating within the central proxy server 12 of Figure 1. These components are preferably part of a central caching module 13 of Figure 1. A data request reception module 102 allows the central proxy server 12 to receive the request for data made by the user at a station 30 with the data request module 92. An Internet request module 104 passes the request on through the Internet 20. A Data reception module 106 receives the data 15 transmitted by the Internet 20 in response. A cache storage module 108 receives the requested data 15 and stores a copy of the data 15 within the master cache database 14. An uplink transmission control module 109 coordinates with the communication network to distribute selected data 15 to the local proxy servers 22. 20 In one embodiment, the uplink transmission control module 109 communicates with the satellite transmitter 16 in transmitting data 15 through the uplink 32 to the satellite 18. Of course, other types of transmission could be utilized, including over the Internet itself.

25 A satellite operation block 110 illustrates the basic operation of functional modules operating within the satellite 18 of Figures 1 and 2. Once the web page 15a has been uplinked by the central proxy server, the satellite 18, an uplink data reception module 112 receives the uplinked web page 15a or other data 15. Subsequently, a data broadcast module 114 formats and transmits the data 15 through satellite multicast 34 to all associated local proxy servers 22. 30

Additionally, if the uplinked data 15 contains private data 55, a pointcast data module 116 formats and pointcasts the private data 55.

An uplink transmitter block 120 of Figure 3 illustrates general functional modules operating within the satellite uplink transmitter 16 of Figures 1 and 2. The data 15 to be uplinked is formatted into a proper form and protocol for satellite transmission with a data formatting module 124. Header information is added to the packets to be uplinked with a header addition module 124. One simplified example of the structure of a packet 190 and header 192 is shown in Figure 4. The uplink transmission is conducted between the satellite uplink transmitter and the satellite 18 with the use of an uplink transmission module 126.

An Internet block 130 illustrates one example of the functional modules operating within the Internet 20 of Figures 1 and 2. Requests for data 15 are transmitted over the Internet 20 with a data request module 132. The request is typically the request generated by the Internet request module 104. Once the data 15 is requested, the Internet site 35 at which the data 15 is hosted is contacted with a site contacting module 134. A site map of the site, including the location and configuration of the web page 15a is transmitted from the web site 35 with a site map transmission module 136. Thereafter, the web page 15a is transmitted from the Internet site 35 to the central proxy server 12 with a data packet transmission module 138. The web page 15a and attendant site map may also be transmitted directly to the requesting local proxy server 22.

A local server block 150 illustrates one embodiment of the functional components operating within the local proxy server 22 of Figures 1 and 2. Preferably, these functional components are contained within the local caching module 17 of Figure 1. Within the local server block 150, a data request block 155 includes a data reception module 152. The data reception module 152 receives requests for Internet data 15 from the end users at the stations 30. Generally, this request is generated by the data request module 92. A search module 154 searches the local cache database 24 for the requested data 15. In one embodiment, the search module 154 comprises the proxy cache protocol employed within the http language.

If the requested data 15 is not present within the local cache database 24, a data request module 156 transmits a request for the data 15 to the central proxy server 12. This request is typically routed over one of the communication channels 38, 40, or 42, through the Internet 20, and through communication channel 36 to the central proxy server 11. The request may comprise the original request received from the user at a station 30 modified with the Internet

URL of the central proxy server 12. Concurrently, the data request module 156 may also request the data 15 directly from the Internet site 35 where the data 15 is hosted.

A data reception module receives the requested data 15. Typically, the data 15 is provided by the central proxy server 12 in accordance with the procedure discussed for the central server module 100. Thus, the data 15 is uplinked 32 to the satellite 18, which multicasts the data 15 to all subscribing local proxy servers 22. The data 15 is received by the satellite receiver 26 and passed on to the local proxy server 22. The local proxy server 22 may also receive the data 15 directly over the Internet. The data 15 from the earliest arriving of the multicast 18 and the direct Internet transmission is then passed on to the user at the station 30.

A data management block 160 illustrates the functional components of the local proxy servers 22 which handle the data 15 received from the satellite transmission 34. When data 15 is requested by a user station 30 and is found to not be present within the attendant local cache database 24, a request is made for the data to the central proxy server 12, as discussed. That data 15 is then requested from the Internet site 35 where it is hosted, as also discussed, and then transmitted through satellite transmission 34 to the local proxy server 22 requesting the data. Concurrently, the data is also received by the satellite receivers 26 of all other subscribing local proxy servers 22 with the use of the data broadcast reception module 162 which is preferably programmed into each of the local proxy servers 22. Typically, all of the satellite receivers 26 will be attuned to the same frequency over which the data 15 is transmitted. Of course, if a number of satellites 18 are used to transmit the data 15 around the world, a number of different frequencies may also be employed.

Once the data 15 is received, a data supervisor module 164 operating within each of the local proxy servers 22 attends to the storage of the data 15 within the local cache database 24. The data 15 is automatically stored for a predetermined amount of time. For example, the predetermined amount of time may be, for instance, a period of four hours. Once that time has passed, A heuristics formula application module 166 employs a heuristic determination to determine whether to continue to store the data 15 or to discard the data. If, after application of the heuristic determination, it is decided that the likelihood of a request for the data 15 is low for that particular local proxy server 22, a push module 168 will discard the data 15, removing it from the local cache database 24. In one preferred embodiment, each local proxy server 22 is provided with its own, individualized, heuristics formula.

In one embodiment, the modules 158, 160 integrally communicate with a cache database management module such as Novell's ICS™ software, which in turn attends to the

storage of the data 15 onto disk at the high rate of speeds specified, and/or Novell's Border Manager™ software which acts as a proxy cache manager and provides proxy content filling and firewall functions.

5 A heuristic determination block 170 illustrates the functional components of one embodiment of a heuristic formula of the present invention. A local statistics collection module 172 resident within each local proxy server 22 monitors the frequency of requests for each file of data 15 located within the local cache database 24. A global statistics reception module 174 receives data demand statistics from a central location. In one embodiment, the central location is the central proxy server 12, which collects demand statistics during the
10 process of servicing requests for data 15. In this manner, the central proxy server 12 acts as a sort of "Nielson Ratings Service" for the Internet. This information may be provided either free or for a fee to interested parties.

A heuristic determination module 176 utilizes the statistical information gathered by modules 172 and 174 in making the heuristic determination of whether to keep a particular file
15 of data 15. If the determination is to keep the data 15, a least recently used (LRU) module may be used to decide which existing data is to be discarded to make room for the new data 15. The LRU calculation module 178 calculates the amount of use of each file of data 15 and adds a local use factor into the determination. This heuristic determination is applied by the heuristics application module 166 as described above.

20 The heuristics determination module 176 preferably utilizes localized web page demand statistics compiled by the local proxy server 22 as well as globalized web page demand statistics collected by and transmitted periodically from the central proxy server through satellite transmission.

A hit rate reporting module 179 periodically reports the local hit rates for part or all
25 of the data 15 transmitted from the central proxy server 12. This information is used to calculate global demand statistics. Preferably, each participating local proxy server 22 tabulates every request from every end user station 30. A hit report is then periodically transmitted to the central proxy server 12. The central proxy server 12 in turn tabulates the accumulated hit reports for all web pages requested and periodically sends updated usage
30 information with the hit totals to the local proxy servers 22. Preferably, every file of data 15 is assigned a globally unique number which can be stored and transmitted compactly and which is used to identify the particular data 15.

It is preferred that the heuristic determination module 170 is closely integrated with the data management module 160. In one embodiment, the data management module 160 is implemented through tight integration with the cache database management module program 29 and the heuristic determination module is configured to work closely with the cache database management module 29.

As one embodiment, the data management module 160 keeps track of related files within a web page 15a or files related to a web page. The related files may be uplinked and transmitted through satellite transmission 34 together. The heuristic determination module 170, being integrated with the data management module 160, may be configured to receive a list of the objects and files associated with a page. In this manner, the data heuristic determination module 170 may be able to ensure that the related data stays together and is stored together in the local cache database 24. It can also calculate usage information for each of the objects and files separately and keep all associated files or "trim" a portion of the objects and files that are not highly used. The related files may then be transmitted as a group when one or more of the related files is requested by a user at a communicating station 30.

Additionally, the heuristics module 170 may also track "supergrouping" of web page information 15. In many web pages, information such as advertisements change or alternate back and forth. These changes may be kept track of and the differing versions transmitted together by the central proxy server 12.

One example of a heuristics determination module 200 is illustrated in Figure 5. Shown therein is a local demand data module 202. In addition, each local proxy server 22 may also have its individualized local policy 204. Within the policy 204 is a relative weighting 206 of local and global web page demand. This policy 204 and the local demand data 202 as well as global demand statistics 208 are employed by the heuristics formula 200 to determine whether to keep or discard each separate data file 15 that is received, once the selected period of time has passed. The local proxy server policies 204 may also weight subject matter of the web pages 15a. The policy 204 may also include other local value factors, such as the ease of getting the data 15 for the particular local proxy server 22.

The policy 204 may also specify the particular interest areas that the heuristics formula is intended to weight. For instance, if the local proxy server 22 services a school, the policy 204 may give high weight to data 15 containing educational issues. If the local

proxy server 22 services a research institution, the policy 204 may give high weight to data 15 relating to the research being conducted highly, etc.

Thus, as shown in a global demand calculation block 180 of Figure 3, a global demand monitoring module 182 resident within the central proxy server 12 monitors all of the requests for information from each local proxy server 22. It also monitors all of the hit rate data transmitted by the reporting modules 179 of each local proxy server 22. In response, a global demand compilation module compiles the data collected by the global demand monitoring module 182. Within those statistics may be information relevant to each particular subject matter such that the heuristics procedures may take the categorized demand data into account according to the particular weighting of the subject categories in their local policies 204.

A particular subscribing local proxy server 22 might be an educational institution, business, or news provider, and might weight subject matter statistics from relevant categories more heavily within its own local policy 204 than others of the local proxy servers 22. A global demand transmission module 186 transmits the demand data, preferably by satellite transmission 34, in the same manner as the requested data 15 is transmitted.

Figure 6 is a schematic flow chart diagram illustrating one manner of operation of a software program 220, a copy of which preferably operates on each subscribing local proxy server 22. In one embodiment, the software program 220 comprises the local caching module 17. The program 220 initializes at a start block 222. The program 220 then proceeds on to a block 224 where it awaits receipt of input to be processed. If the input is a terminate command, the program 220 receives the command at a block 226 and then progresses to an end block 228, where the program 220 terminates.

Three representative procedures are shown in Figure 6, corresponding to several different general functions that may be accomplished by the program 220. For instance, if the program 220 receives a user request for data 15, the program receives the request at a block 230. The program 220 then proceeds to a block 232, where the caching protocol of the http language is utilized to consult the local cache database 24 and thereby determine if the requested data 15 is present. At a query block 234, the program 220 branches in one of two directions depending on whether the data 15 is present in the local cache database 24. If the data is present, the program proceeds to a block 236 where it transmits the data 15 to the requesting station 30 for presentation to the user. The program then proceeds to a block 238, which returns control of the program to the start block 222.

If the data 15 is not present, the program 220 proceeds from the query block 234 to a block 240, where it passes the request on through the Internet 20 to the hosting site 35. At about the same time, the program 220, at a block 242, transmits a copy of the request to the central proxy server 12. The program 220 then waits at a block 244 for the requested data 15 to return by way of the fastest route, whether it be through satellite transmission 34 or through the Internet 20. Once the data 15 is received, the program 220 transmits the data 15 to the requesting user station 30 for presentation to the user. Thereafter, the program 220 moves to a block 247 where the program 220 updates local demand statistics. Subsequently, the program moves to a block 248 which passes control back to the receive input block 224.

If the input received by the receive input block 224 is a the transmission of web pages 15a or other data 15 by transmission 34, the program moves to a block 250, where the transmitted data 15 is received. The program 220 then proceeds to a block 252 where it stores the received data 15 in the local cache database 24 for a selected amount of time. After the selected amount of time has passed, the program 220 generates heuristic data utilizing the heuristic formula 200 of Figure 5, and makes a determination of whether to retain or discard the data 15.

This heuristic determination is made substantially in the manner described above in relation to the discussion of Figure 5. Thereafter, the data 15 is either stored for a greater period of time or is discarded at a block 256. If the data 15 is stored and the local cache database 24 is currently fully filled with data 15, a LRU procedure (e.g., module 178 of Figure 3) is used to discard the least recently used data 15 within the database 24 or the data for which the LRU procedure otherwise determines is the least likely to be requested in the future. Thereafter, the program 220 progresses to a block 258 which returns control to the start block 222.

If the input received by the receive input block 225 is global demand statistics from the central proxy server 12, the program 220 progresses to a block 260 where the global demand statistics are received. Subsequently, at a block 262, the local record of global demand statistics is updated. At a block 264, the local use statistics generated at block 247 are consulted and obtained. The local use statistics are compiled together with the global demand statistics at a block 266. The heuristic procedure 200 of Figure 5 is then employed at a block 268 to calculate the local hit potential of the particular file of data 15. The hit potential is then compiled as heuristics data for use in block 254 (and module 166 of Figure 3). The heuristics data is then stored and copies are passed on to the modules which must employ the heuristics

data to make a heuristics determination at a block 272. The program 220 then proceeds to a block 274 which passes control back to the receive input block 224.

Figure 7 is a flow chart diagram illustrating one manner of operation of a software program 300 resident within and operating on the central proxy server 12 of Figures 1 and 2.

5 The program 300 initializes at a start block 302 and proceeds to a block 304, where it awaits receipt of a request for data 15. Of course, the program 300 could have other means of identifying high demand data 15. Web site search engines or other pre-generated statistical data could be consulted, for instance. The central proxy server 12 could also keep a history of data requests for web pages 15a, which are updated daily or weekly, etc. in order to re-
10 transmit those pages 15a once updated.

Once data 15 is requested by a local proxy server 22 or otherwise identified as being of sufficient demand to merit caching within the subscribing local proxy servers' attendant cache databases 24, the program 300 proceeds to a block 305. Here, the program 300 requests a copy of the data 15 from the hosting Internet site 35. Subsequently, as designated by a block
15 306, the data 15 is received over the Internet 20 from the hosting site 35. The data 15 may then be filtered before being transmitted and/or stored within the master cache database 312. For instance, a morality filter could be used to filter out obscene language or pornographic materials. Other types of filters could also be used by content, or filters could be used to filter out data for which it is known that hit rates in the future will be low. A heuristic scheme or
20 some type of request history might be employed for so doing.

At a block 310, the data 15 which has successfully passed through filtering is stored in the master cache database 14 as represented at a block 210. The data 15 is then sent to the broadcast uplink transmitter 16 as represented by a block 312. As discussed, the
25 communication medium may be any suitable medium, but satellite transmission will be discussed herein. As shown by a block 314, the data 15 is then transmitted to the satellite 18. Thereafter, as shown by a block 316, the data 15 is transmitted by satellite transmission 34 to the subscribing local proxy servers 22. This transmission may be coordinated by the program 300.

At a subsequent block 318, global demand statistics are updated to reflect the request
30 for the data 15. The updated global demand statistics may be transmitted at this point to the local proxy servers, or this information may be periodically transmitted. Subsequently, the program 300 progresses to a block 322 which returns control to the start block 302.

The system 10 of the present invention moves high demand web pages to the edge of the Internet, closer to users. The result is a much faster delivery of a majority of web pages requested by end users at a station 30, and a corresponding decrease in Internet traffic, substantially accelerating the Internet. Connection costs will correspondingly decrease, resulting in savings to users employing the system 10.

Example of Central Proxy Server Operation

Referring to Figure 8, shown therein is one representative example of the operation of the central proxy server 12 of Figure 1. Within Figure 8 a number of local proxy servers 22 are shown communicating with a central proxy server 12 via a communications/validation manager 332. The communications/validation manager 332 maintains a secure connection from the central proxy server 12 to each of the local proxy servers 22 in the system. Messages from the local proxy servers 22 are communicated through the message switcher 338 to the appropriate module.

One of the primary modules for handling messages is the miss/refresh handler 340. When a local proxy server 22 is missing any requested HTML data 15a or any other requested global communications network data 15, a miss is then passed onto the central proxy server 12 and received by the miss handler 340. The miss handler 340 then goes out to the cache database management module 29, represented herein as an Internet Caching System (ICS), one example of which is the ICS program available from Novell Corp. of Provo Utah, and requests the data associated with that miss. The cache database management module of the central proxy server will either have that data 15 locally in the local cache database 22 or it will go out to the Internet 20 and request that data 15.

When that data is received, all the information needed to assemble that information in that particular HTML page or other data within our system is present. The complete set of data 15 within a web page 15a is then passed on to a priority queue 342. The priority queue 342 handles all of the prioritization of various HTML pages or data for later transmission. When a particular file of HTML data has a sufficiently high priority, it is placed into the transmission queue 342 which can then request all of the data associated with this HTML page, queue it up for transmission, and send it out to the packetizer 348, where it is placed in full packets and sent off to the communication medium (in one embodiment, the satellite 18).

The central proxy server 12 preferably also includes a control sub-system 350 which handles all of the start-up, shutdown, initialization and processing. An attention sub-system

352 is also preferably provided and handles all user interface types of interaction. The central proxy server 12 is preferably in communication with a database 334 which may employ a schema for handling historical logging of information transmitted back from the local proxy servers 22 to the database 334 upon a message that is sent by the central proxy server 12.

5 The web data 15 which may be handled and forwarded through the system include web objects or groups of web objects that form a web page 15a. Web objects are also referred to as HTML objects or HTML data.

 When the data 15 is placed on the priority queue, the priority of that data is altered somewhat continuously as the central proxy server 12 receives indication from the local proxy servers 22 that one or more end users have made a cache hit on that object or that a local proxy server 22 reported another miss. This popularity information keeps accumulating even after the initial insertion into the priority queue and the highest priority data files 15 are considered for transmission.

 The transmission queue 346 draws the highest priority data files 15 off the priority queue and then attempts to gather together the web objects, container objects, and their component objects and form that data 15 into logical messages. The input side of the transmission queue 346 receives the data 15. The output of the transmission queue 346 is a list or multiple lists of logical messages that represent pieces of that data 15.

 The transmission queue 346 processes the data 15 and formulates logical messages, all of which are preferably small, such as about 8 Kbytes. These logical messages are placed onto an output queue, or multiple output queues, and that data serves as input to the packetizer module 348.

 The responsibility of the packetizer module 348 is to take those logical messages, whatever size they are, and make them fit into IP multicast packets in such a way that there is no wasted space. In other words, if there are several small logical messages, they would get packed one end-to-end inside of those IP multicast packets.

 The maximum packet size is preferably selected to stay within the maximum data payload of an Internet frame, about 1500 bytes. But more importantly, the length of the data in each of the packets is preferably optimized to be approximately 1442 bytes. That number is selected in one embodiment to work with the subsequent transmission of that packet over a transmission medium such as a satellite system which incorporates DVB packets and the multi-protocol encapsulation or MPE encodings. The size is chosen to be optimal for data

efficiency, so that there is no wasted bandwidth on the satellite signal. Thus, there are no fragmentary DVB (Digital Video Broadcast) packets, in one embodiment.

Once the packetizer module 348 has assembled the logical messages and filled up a packet, it will forward that packet to the satellite uplink facility 16, which in this embodiment
5 comprises an IP encapsulator. The IP encapsulator takes packets using the IP encapsulation and translates them into bits and pieces that match the DVB packet sizes, which are much smaller.

In fact, each DVB packet may hold only approximately 188 bytes of data. The total size of the packet being 204. Some of the packets may be occupied by header information and error correction information. The IP encapsulator outputs those DVB packets ready to go, and
10 they then pass through standardized hardware to for radio frequency conversions and the like, as is commercially available and well known.

The packets in this embodiment are then sent from the transmission dish up to geosynchronous orbit, hit the satellite, bounce off it, come back down, and are received at each of
15 the subscribing local proxy servers 22. Preferably, a satellite receiver at each site receives and recombines the DVB packets into the original IP multicast packets that were fed into the IP encapsulator at the uplink facility.

The IP multicast is a standard protocol known in the industry. The packet structure of IP multicast is the same as any IP packet used with TCP/IP or UDB/IP transport protocols.
20 The only difference is the addressing of where it is going to, i.e. the target addressing uses special range IP addresses which indicate lots of people can receive the same thing at once.

Under this embodiment, a one to many transmission originates at the central proxy server and is delivered to the local proxy servers 22. The satellite receiver or other transmission medium receiver reconstructs and outputs the original IP multicast packet onto a
25 local Ethernet network, which is preferably a private network, and may be a single cable between the satellite receiver and the user net in the local proxy cache machine. The local caching module 17 receives a payload from those packets and is processed as illustrated in Figure 10.

Referring now to Figure 9, shown therein is a block diagram illustrating one
30 embodiment of the configuration of packets constructed in the transmitting of data 15. Initially shown is the transformation of the web objects 15 that are received out of the transmission queue 346 of Figure 8 and inserted into individual object messages 360.

Each data file 15 is preferably broken up into multiple object messages 360 from 1 to N. Each object message 360 consists of an object message header 362 and a payload 364. Each object message 360 is then broken up into a satellite packet 366 by the packetizer module 348. The Satellite packets, as stated, are 1442 bytes in size or may be another suitable size selected to give the optimal satellite throughput.

The satellite packet similarly consists of a packet header 368 and a payload 370. There are 1 to M packets for every given object message. The satellite packets are then converted into IP multicast format via industry standard operations. The IP multicast is then converted into a digital video broadcast (DVB) stream, which is once again the industry standard.

The DVB packets can be constructed using a hardware and software solution. An example of this is a DVB packetizer product which is provided by Skystream International. On the receive side, the DVB packets are received by a satellite modem, which then rebuild those packets into IP multicast format. The thusly formatted packets are then received by the local proxy server 22, reassembled, and eventually the entire web object 15 is reconstructed using the inverse of the process previously described.

Example of Local Proxy Server Operation

Referring now to Figure 10, shown therein is a block diagram illustrating the operation of one embodiment of a local caching module 17 of a local proxy server 22 of Figure 1.

Within Figure 10 is shown a satellite receiver 372. Emanating from the satellite receiver 372 are transmission signals 374 and 376 indicating network transmissions using multicast packets for Internet data files 15. The data files 15 are shown being received by a receive assembler module 378.

Each depicted module represents a software subsystem or component of the local caching module 17. A receive assembler 378 receives the IP multicast packets and as it receives one packet and then the next packet and so on, as fast as it is able to, it reconstructs the object messages, which represent portions of data files 15.

The data files 15 are transmitted via an message dispatcher 380, which is basically a switching mechanism, to an injector module 392. The injector module 392 accumulates the object messages, one or more object messages per web object, reassembles those messages into the web object, and injects that web object data 15 into the local cache database management module 29.

The injector module 392 preferably not wait until all of the object messages for a data file 15 have been received. As long as they are coming in order and it has the very first message it will proceed to create that object as far as the cache database management module is concerned and will continue to append the additional data as it arrives until the last has arrived. If anything is out of order, it waits until it receives the missing portions.

A satellite health manager 386 queries the satellite receiver 372 about every fifteen seconds sending SNMP (Simple Network Management Protocol) queries by using the UDP/IP transport and the satellite receiver, if all is well, responds affirmatively.

If that is not the case, satellite health manager 386 sends a message via the message dispatcher 380 to a central proxy server session manager 382. The session manager communicates with the central proxy server 12 over a TCP/IP connection 384. That connection is preferably maintained over the standard Internet and the report of satellite receiver misbehavior can thus reach the central caching management module 13 and trigger an alarm or notification such that operations personnel can try to identify the problem and resolve it.

Simultaneously the satellite health manager 386 logs an error in the systems error log on the local caching module 17. An attention subsystem 402 is also preferably provided. Therein is shown a GUI information module 404 an error log status module 406 and a statistics module 408. The attention subsystem 402 is responsible for bringing attention to things by displaying information through the graphical user interface to the customer or to any remote administrator that is hooked into the cache database management module and accessing the specific management panels, as well as logging those errors in a local file.

A remote console 388 is provided for diagnostic purposes. The present invention allows a remote connection over the regular Internet to the local caching module 17 with a simple utility which implements what appears to be just a simple command line such as a DOS command line. The system may enter commands of its own through the local caching module 17 that will give back responses providing information about internal statistics and performance and other diagnostic information and those commands may also be used to set or change internal parameters or characteristics of the local caching module 17 for purposes of optimizing its performance.

A client browser 25 may be browsing the Internet through that particular local cache database management module box. If that client requests a web object which has not been previously cached by the local cache database management module, then the local cache

database management module will call up the miss manager subsystem 400 and identify the URL (universal resource locator) of that web object on the Internet and give the local caching module 17 an opportunity to take steps to find and gather that web object data and deliver it to the local cache database management module 29 by means of the present system, which is
5 what has been described up to this point, where the data is received over the satellite by the local caching module 17 from the Central proxy server and injected or discarded back into the local cache database management module. So the local cache database management module is given first right of refusal and will try to satisfy the end user request for a particular web option.

10

Example of Hit and Miss Rate Reporting

Referring to Figure 10, the local cache database management module 29 is preferably configured to report cache misses to the miss manager 400 of the local caching module 17. In response, the miss manager 400 preferably then transmits a message immediately to the
15 central proxy server 12 via the message dispatcher 380 and the central proxy server session manager 382 to instruct the central proxy server 12 that the local proxy server 22 does not contain a copy of the web object 15 the end user is looking for.

20

The central proxy server 12 then preferably uses the miss information in calculating global miss rate information. That is, the central proxy server 12 uses the miss reports that it receives from the local proxy servers 22 in the overall system 10 to compute the relative priority of web objects 15 in the priority queue 342.

25

Referring back to the local caching module 17 of Figure 10, also provided therein in the depicted embodiment is a hit manager module 396. The hit manager module 396 is similar to the miss manager 400 in that when a client requests a web object 15 from a local cache database management module 29 and the cache database management module 29 has that object already stored in its cache, a hit is registered. The local proxy server 17
accumulates hit counts on an object-by-object basis as they are reported. It then reports that information at least every few seconds to the central proxy server 12 advisor via the message dispatcher and the central proxy server session manager.

30

The local caching module 17 preferably makes those hit reports under several different circumstances. Initially, it may only have room to track hits on so many objects at once, and that determines somewhat the size of the report messages that it is going to send to the central

proxy server 12. If the message contents are full, then it sends the message allowing it to empty or zero out all of those accumulating buffers, counters, accumulators.

Each of the web objects that the system 10 has delivered is preferably assigned a globally unique identifier and the hit count report utilizes that identifier to be very efficient in forwarding those hits to the central proxy server 12. In so doing, it may simply send out an array of structures. Technically, the structure is an array comprised of two elements. One element is the global ID which represents the web object that the system 10 is tracking. The other element is a count of the number of hits that has just been recorded with a very recent task.

As that array of structures fills up or is used up by recording hits for different objects, the current message is sent as soon as all of the array structures are in use or, if not all of them are in use, the partial messages are transmitted no greater than about five seconds apart to make sure that information stays timely until it gets back to the central proxy server 12. That array of information or partial array is also transmitted immediately if one or more of the objects which are being hit has experienced a very large number of hits in a very short period of time. A type of frequency measurement is observed, as well as just accumulations of total hits.

All of these messages that are sent between the central caching module 13 and local caching module 17, whether they are sent over the traditional Internet using TCP/IP connections or whether they are delivered via the satellite transmission, share one common protocol structure or proprietary protocol means. At the beginning of the message, there is a header that identifies the overall length of the message and the target of the message.

For example, when a miss manager of a local caching module 17 needs to send a miss report, it will prepare a message and a header will identify the target as "central proxy server subsystem miss handler." When the central proxy server 12 transmits object messages, the header of those messages will target to the local caching module 17's injector subsystem and it is the responsibility of the message dispatcher box to switch those messages back and forth between the appropriate subsystems on the local or master.

The present invention thus reports on hits that have been registered on objects that are being tracked by the system 10 and attempts to report those hit counts as quickly and efficiently as possible so that they have relevance. When the hit report reaches the central proxy server 12, those hit counts are matched up with the appropriate web object that is being tracked using the global ID. The hit counts are then added into the prior known hit counts for

that object and a new priority for that object can be calculated using miscounts and hit counts to date and some other information.

5 Every time a hit or miss report is received by the central proxy server 12, it will accumulate that and update the totals, but it may not actually transmit, or may not complete the recalculation of the priority because doing so requires removing the object from the priority queue and reinserting it at a new position. Given that there will likely be hundreds of thousands of web objects represented from that priority queue, that becomes an expensive proposition in terms of computer processing time. So instead the potential change in the priority are evaluated and when it crosses a certain threshold or every so many times a report 10 has been received, then the object is reinserted into its appropriate position.

Relative priorities of any particular object are preferably transmitted under two circumstances. When the object data itself is transmitted, the priority of that object is transmitted with it. That information is provided to the local caching module 17 and to the local cache database management module. The local cache database management module at 15 that point preferably incorporates that global hit data that has been provided into its own internal calculations of how wonderful this object is and how long it wants to keep it around in its local cache. The second time that an object's priority is typically transmitted is when that priority has changed noticeably, somewhat substantially. But the object data itself would not change. There would be no need to resend the subject data but it is useful to send the very 20 brief, very small administrator message saying the object identified by global ID such and such has a new global priority, so please consider it.

In addition to recording hit counts on a moment to moment basis for various objects, the hit manager preferably also keeps the longer term total of those hits. In fact, it preferably keeps a total for a time period of, for example, fifteen minutes. Every object that has had any 25 hits on it in 15 minutes, at the end of that 15 minutes is then evaluated relative to all the others. The objects are sorted out and the record of the ordered top 100, 200 or some selected number of objects are logged to a file in the local cache box and that file will be available for subsequent retrieval by the central caching management module 13 and the historical database agent

30 The DB agent goes out periodically and queries each of these local caching modules 17 getting all of the accumulated 15 minute logs and data regarding what was most popular of your local cache during that 15 minute, and that will update all the different local caching modules 17 and will present a number of opportunities for data mining.

One advantage of the system 10 of the present invention is that there exists a closely integrated relationship between the local caching module 17 and the cache database management module 29. This allows the system 10 to communicate information such as the hit reports that is not available in prior art caching systems. That information may be gathered even after the object has unloaded into that cache and it can be monitored subsequently.

Another advantage is that hit counts are accumulated within hit reports at the local cache. Many hit counts may be accumulated before transmission due to time constraints. Thus, a number of hits on a selected object may be transmitted in a single report.

Example of Tightly Integrated Caching Software

Referring to Figure 10, a number of arrows are shown directed into and out of an cache database management module 29. These arrows in essence represent the functional interface for an application programming interface (API). In one embodiment, the tight integration between the cache database management module 29 (one example of which is the cache database management module of Figure 10) and the local caching module 17 is conducted through the use of an API. The term "API" as understood in the industry is a collection of programming interfaces that allow one set of software to access services or information from another set of software in a standardized manner which separates and protects each side from the other, through establishing a common method of communication. Typically, the two software modules operate on a common server and their communications are transmitted directly between each other.

The API can be considered a type of dynamic binding between two applications operating on a common server or other processor. The terms "integral communication," "integrally communicate" and "tight interface" represent some sort of binding between two separate applications on a common processor. As disclosed, the binding is preferably a dynamic binding, one example of which is conducted through an API.

An API defines all of the interface information that must be implemented by a vendor in a product. The use of an API-type interface in one embodiment allows the present invention to benefit from close, substantially instantaneous communication between the local caching module 17 and the cache database management module 29. For example, referring to the injector module 392 of Figure 10, arrows pass from the injector module 393 into and out of the cache database management module 29. In one embodiment, four functions are provided for the injector to call from within the cache database management module 29. Each

of those functions is called with a short command, and if necessary, accompanying parameters. Those commands are responded to with the performance of the requested function, which may, for instance, include the depositing of data in a selected buffer or memory location.

5 As an additional example, when the injector module 392 receives web objects or other files 15 and needs to transmit those objects to the cache database management module 29, it uses commands from selected API instruction sets and passes those commands to the cache database management module 29 to achieve that desired function.

10 Other API functions may include an instruction or command which requests the cache database management module 29 to locate an object 15 within the local cache database 24. Another API instruction may notify the cache database management module 29 that a particular data file 15 is being sent and is to be saved together with another stored data file 15. This data may not actually be part of the original object data 15, but may be related information, such as a component of a web page 15a.

15 In one embodiment, the cache database management module 29 may respond to the instruction asynchronously, that is, at varying times. The responses may be immediate, or may be delayed, to acknowledge the results of an initial request, for instance. Other functions may comprise cache database management module one-way communication to the local caching module 17 to provide notification of certain events such as a hit report or a miss report.

20 Thus, an API interface is very highly integrated between the local caching module 17 and the cache database management module 29. This provides the advantages that the cache database management module 29 can be communicated with internally and unique types of information can be received back from the cache database management module, including statistics such as hit reports, almost instantaneously.

25 Integration does require some additional support and alteration by the vendor and produces the results of greater flexibility of management and information extraction. Additionally, speed is increased, so that reported information is more current. In prior art arrangements, where an software applications residing with different external servers
30 communicate, there are necessarily slowdowns resulting from network transmission delays, processing of network messages in and out of the boxes, and slowdowns resulting from hardware bottlenecks. These include the use of drivers of the hardware and slow funneling of information that must work its way up through the various hardware protocols and then back

down again. These slowdowns are avoided through the tight integration of the present invention. Communication may be as simple as making and responding to a function call.

Prior art caching system typically receive new objects in whatever order they are sent to that caching system. The caching system then just throws out the least recently used data to make way for the new data coming in a very unintelligent manner. Prior art systems may provide limited feedback on misses, due to the need to request data that is not present in the cache. In such systems, there is no sense of global priority. That is, each remote caching system is unaware of what is popular elsewhere, other than a possibly a rudimentary calculation based upon misses. The prior art caching box responds to transmitted data by makings to make room for this new arrival, and use it's own best judgement.

Thus, one embodiment of the present invention is based directly on having an integrated API relationship between the cache advisor and the cache software. This allows the system 10 to have access in real time to prepare information about what is going on inside the cache database management module 29, and also what is going on relative to particular web objects 15 that are being tracked. Given that information, the local caching module 17 and the central proxy server 12 collaborate. Each of the local caching modules 17, that are online and subscribed to the system 10, report their popularity information to the central caching management module 13, which compiles that information and sends it out with extracted data files 15. The local caching modules 17 then utilize that global popularity data in determining what data 15 to keep and what to discard, optionally in combination with unique local determination factors.

Local proxy servers 22 thus act as the eyes and ears across the world. Due to the ability to communicate closely with the cache database management module, the local proxy servers 22 are able to extract at least hits, and optionally, many other types of information from the local cache databases 24. Examples of information that can be extracted from the local caching module 17 include the acquisition time -- the time that it takes to fill an object over the Internet; information such as what is being deleted out of the local cache, which may indicate local priorities; time data -- data that is hot in a particular area or time zone; hit time -- the time it took the local cache database management module to retrieve an object from the Internet; and other particularized information about what objects are being kept or thrown out. Even the time zone of where that local cache is can be related and associated with the global popularity data.

Transmission of Web Pages in Their Entirety

The system 10 of the present invention allows a web page to be transmitted in its entirety, rather than transmitting a listing of the contents of the page and going back separately requesting those contents as is conducted with most Internet accesses. One feature preferably
5 provided within the cache database management module 29 of the present invention is the ability to examine an HTML web object (such as a web page) 15 and determine its component parts. The HTML object preferably includes a container that describes the complete layout of a particular page 15a that will be seen on the browser window 25 of a user 30. The container or other such directories typically contain most of the readable text and contain specific
10 instructions, using the hypertext mark up language syntax, to identify other web objects, images, Java applets, or other types of web object that are part of the page.

The cache database management module also preferably looks at a potential container that refers to all of the other images and elements that comprise the total Web page experience. The cache database management module 29 will, upon receiving the HTML
15 object 15, scan through it immediately, parsing out the particular references to other objects that are components of that page. As the cache database management module 29 finds each of those, it will initiate a request. The system 10 first checks to see if the image is already present in the local cache database 24. If it is not, the system 10 initiates a request over the Internet 20 to retrieve that image and continues to do that for every object that was indicated as a component of that HTML container object.

This feature provides an additional acceleration to the end user 30. The cache database management module 29 has hopefully retrieved some, if not all, of the component objects that complete a web page by the time that the end user browser 25 has received the original HTML content. The cache database management module 29 preferably begins to analyze and parse
25 the requested object 15 and goes back to the cache database management module 29 requests all component objects of the web page 15. The entire contents of a web page 15a are then streamed to the web browser 25, without waiting for the web browser 25 to request them individually. Without this read-ahead feature, as it is called, on the cache database management module 29, the browser 25 after receiving the original HTML object, would then
30 have to request each of the component objects with a separate request, and each of those may result in cache request, and subsequently, a request across the Internet. Whereas with the read ahead feature of the cache database management module, hopefully many of those objects 15 are already in the cache database management module and the client browser 25 will receive

them automatically or alternatively receive immediate responses when the browser 25 requests those objects 15.

5 This "read-ahead" feature is preferably also provided within the central caching management module 13. Accordingly, when a cache miss request arrives from a local caching module 17, and the central caching management module 13 is requested to retrieve this web object 15, that web object will generally be the initial reference to a particular web page 15. Thus, the object 15 is preferably retrieved by the cache database management module of the central proxy server 12 and handed off to the central caching management module 13 along with some identifying information that the received object is an HTML container that may
10 have certain components. The cache database management module 29 of the central proxy server 12 then preferably automatically pre-fetches or pre-locates those components, whether in its local cache or out on the Internet and notifies the central caching management module 13 as each one of those are identified and found. This allows the central proxy server to be aware of the entire structure of a web page early on.

15 In one embodiment, the HTML object, together with a list of component objects that accompany the web page 15a are stored together in the cache database management module 29, both of the local proxy servers 22 and the central proxy server 12. The system 10 knows about the HTML object and when it transmits the content of that web page 15, it not only sends the HTML object, but it preferably also sends, together therewith, data for all of the
20 component objects that it knows about. When that information arrives, it is passed through to the injector subsystem. Part of those messages will be the container, the information that identifies that particular HTML object and represents the whole page. The container preferably also identifies which of the objects that are being received are component objects of the page 15a.

25 With that information about which objects comprise a web page, the injector subsystem can begin to inject all of those web objects into the local cache, and when it has completed putting all of those objects in, it can then inform the local cache 24 of that relationship. In certain embodiments, the grouping information may come first or it may come last, but the concept is that the local caching module 17 has the opportunity to inform
30 the local cache that these objects all go together and form a logical entity called a web page. This information is preferably used by the local cache to optimize its storage of those objects on the hard disk to keep them in close proximity for later retrieval if need be, so that retrieval time is cut to a minimum.

CLAIMS:

1. A system for accelerating distribution of information over a global communications network, the system comprising:

5 a central proxy server configured to extract data files from a global communications network and transmit the extracted data files over a communication medium;

a local proxy server configured to receive the extracted data files from the central proxy server over the communication medium, provide the extracted data files to a caching database in local communication with the local proxy server, and communicate with a plurality of customer stations to receive requests for information available at remote locations across the global communications network;

10 a cache database management module configured to communicate with the caching database in order to store the extracted data files within the caching database and to retrieve the extracted data files from the caching database; and

15 a local caching module operating within the local proxy server and in integral communication with the cache database management module, the local caching module configured to receive the extracted data files from the communication medium and instruct the cache database management module to store at least a portion of the extracted data files within the caching database,

20 the local proxy server also configured to respond to the requests from the customer stations by the cache database management module checking whether the requested information is among the extracted data files stored within the caching database, and if the information is not among the extracted data files stored within the caching database integrally communicating with the local caching module that the information is not present.

25 2. The system of claim 1, further comprising an application program interface (API), the local caching module in integral communication with the cache database management module through the API.

30 3. The system of claim 1, wherein the cache database management module is a commercially available Internet caching system program.

4. The system of claim 1, wherein the local caching module is configured to, through the integral communication with the cache database management module, receive

updates of the hits and misses of selected data files which customer stations have requested from the local proxy server and transmit that information to the central proxy server for use in selecting which data files to extract from the global communications network.

5 5. The system of claim 4, wherein the cache database management module is configured to receive requests for the updates of the hits and misses, and the local caching module is configured to request the updates of the hits and misses from the cache database management module.

 6. The system of claim 1, wherein the communication medium comprises an IP multicast system.

10 7. The system of claim 1, wherein the communication system comprises multicast distribution of the data files through geo-synchronous satellite transmission.

 8. The system of claim 1, wherein the communication medium comprises geo-synchronous satellite transmission.

 9. The system of claim 1, wherein the local caching module is further configured
15 to respond to requests from a customer station for information available at a remote location on the global communication network by determining whether the information is present in the caching database and if the information is present, transmit the requested information to the customer station and if the information is not present, transmit a notification to the central proxy server that the information was requested by a customer station.

20 10. The system of claim 1, wherein the local caching module is a software program operating primarily in an IPX protocol native to the local proxy server.

 11. The system of claim 1, wherein the central proxy server is further configured to receive updates from a plurality of communicating local proxy servers regarding the number of requests for data files residing at remote locations on the global communication network,
25 compile the updates to determine the relative demand for the data files and transmit the most popular data files concurrently to a plurality of communicating local proxy servers over the communication medium.

 12. The system of claim 11, wherein the local proxy server is further configured with a heuristic procedure for determining whether to retain the extracted data files in the
30 caching database or to discard the extracted data files.

 13. The system of claim 12, wherein the heuristic procedure employs a policy for assigning different weights to different types of extracted data files in making the

determination whether to retain the data files, the policy being unique to the local proxy server.

14. The system of claim 12, wherein the local proxy server is configured to receive from the central proxy server global statistical usage data regarding data files previously transmitted from the central proxy server to the local proxy server.

15. The system of claim 14, wherein the heuristic procedure includes consideration of local statistical usage data regarding the frequency of requests for specific data files together with consideration of the global statistical usage data.

16. The system of claim 12, wherein the heuristic procedure accounts for a particularized subject matter of interest to users of the local proxy server in making the determination whether to retain the data files, and wherein the global statistical usage data includes data indicative of the demand for data files specific to the particularized subject matter.

17. The system of claim 1, wherein the central proxy server is configured to transmit extracted data files to a plurality of local proxy servers, and wherein private data files are transmitted in addition to global data files, the private data files being received by subscribing local proxy servers to a private network and not received by the others of the plurality of local proxy servers which are not subscribers to the private network.

18. The system of claim 1, wherein the central proxy server is further configured to assign all extracted data pages a unique identifier code and transmit the unique identifier codes over the communication medium together with the extracted data pages.

19. The system of claim 18, wherein the central proxy server is further configured to ascertain a popularity rating of each extracted data file and correlate a popularity rating of each extracted data file with the unique identifier code for that extracted data file.

20. The system of claim 18, wherein at least a portion of the extracted data files comprise web pages having multiple components normally transmitted separately, and further comprising associating the multiple components of a web page together such that when a customer station requests the web page, the multiple components of the web page are transmitted to the customer together.

21. A method for accelerating distribution of information over a global communication network, the method comprising:

extracting data files from a global communications network;
transmitting the extracted data files over a communication medium;

receiving the extracted data files over the communication medium into a local proxy server; and

integrally communicating with a cache database management module to store the extracted data files in a caching database;

5 receiving a request from a customer station for information available at a remote location on the global communications network;

integrally communicating with the cache database management module to check for the information among the extracted data files and making the information available for forwarding to the customer station if present among the extracted data
10 files.

22. The method of claim 21, wherein integrally communicating with the cache database management module comprises communicating through an application program interface (API).

23. The method of claim 21, wherein the cache database management module is a
15 commercially available Internet caching method program.

24. The method of claim 21, further comprising integrally communicating between the local proxy server and the cache database management module to receive updates of the hits and misses of selected data files which customer stations have requested from the local proxy server and transmit the updates to a central proxy server for use in selecting which data
20 files to extract from the global communications network.

25. The method of claim 24, wherein the cache database management module is configured to receive requests for the updates of the hits and misses, and the local caching module is configured to request the updates of the hits and misses from the cache database management module.

25 26. The method of claim 21, wherein transmitting the extracted data files over a communication medium comprises transmitting the extracted data files through IP multicast.

27. The method of claim 21, wherein transmitting the extracted data files over a communication medium comprises transmitting the extracted data files through geo-
30 synchronous satellite transmission.

28. The method of claim 21, wherein transmitting the extracted data files over a communication medium comprises transmitting the extracted data files through geo-synchronous satellite transmission.

29. The method of claim 21, further comprising responding to requests from a customer station for information available at a remote location on the global communication network by determining whether the information is present in the caching database and if the information is present, transmitting the selected data file to the customer station and if the information is not present, transmitting a notification to a central proxy server that the information was requested by a customer station.

30. The method of claim 21, further comprising operating the local caching module primarily in an IPX protocol native to the local proxy server.

31. The method of claim 21, further comprising receiving into the central proxy server updates from a plurality of communicating local proxy servers regarding the number of requests for data files residing at remote locations on the global communication network, compiling the updates to determine the relative demand for the data files and transmitting the most popular data files concurrently to a plurality of communicating local proxy servers over the communication medium.

32. The method claim 21, further comprising conducting a heuristic determination of whether to retain the extracted data files in the caching database or to discard the extracted data files.

33. The method of claim 32, wherein conducting a heuristic determination further comprises assigning a policy of different weights to different types of extracted data files, the policy unique to the local proxy server.

34. The method of claim 32, further comprising receiving from a central proxy server global statistical usage data regarding data files previously transmitted from the central proxy server to the local proxy server.

35. The method of claim 34, wherein conducting a heuristic determination further comprises considering local statistical usage data regarding the frequency of requests for specific data files together with considering the global statistical usage data.

36. The method of claim 32, wherein conducting a heuristic determination further comprises accounting for a particularized subject matter of interest to users of the local proxy server in making the determination whether to retain the data files, and wherein receiving the global statistical usage data includes receiving data indicative of the demand for data files specific to the particularized subject matter.

37. The method of claim 21, further comprising transmitting extracted data files to a plurality of local proxy servers over the communication medium and transmitting, in

addition to the extracted data files, private data files, that are received only by subscribing local proxy servers to a private network.

38. The method of claim 21, further comprising assigning all extracted data files a unique identifier code and transmitting the unique identifier code over the communication
5 medium together with the extracted data file.

39. The method of claim 38, further comprising ascertaining a popularity rating of each extracted data file and correlating a popularity rating of each extracted data file with the unique identifier code for that extracted data file.

40. The method of claim 38, wherein at least a portion of the extracted data files
10 comprise web pages having multiple components normally transmitted separately, and further comprising associating the multiple components of a web page together such that when a customer station requests the web page, the multiple components of the web page are transmitted to the customer together.

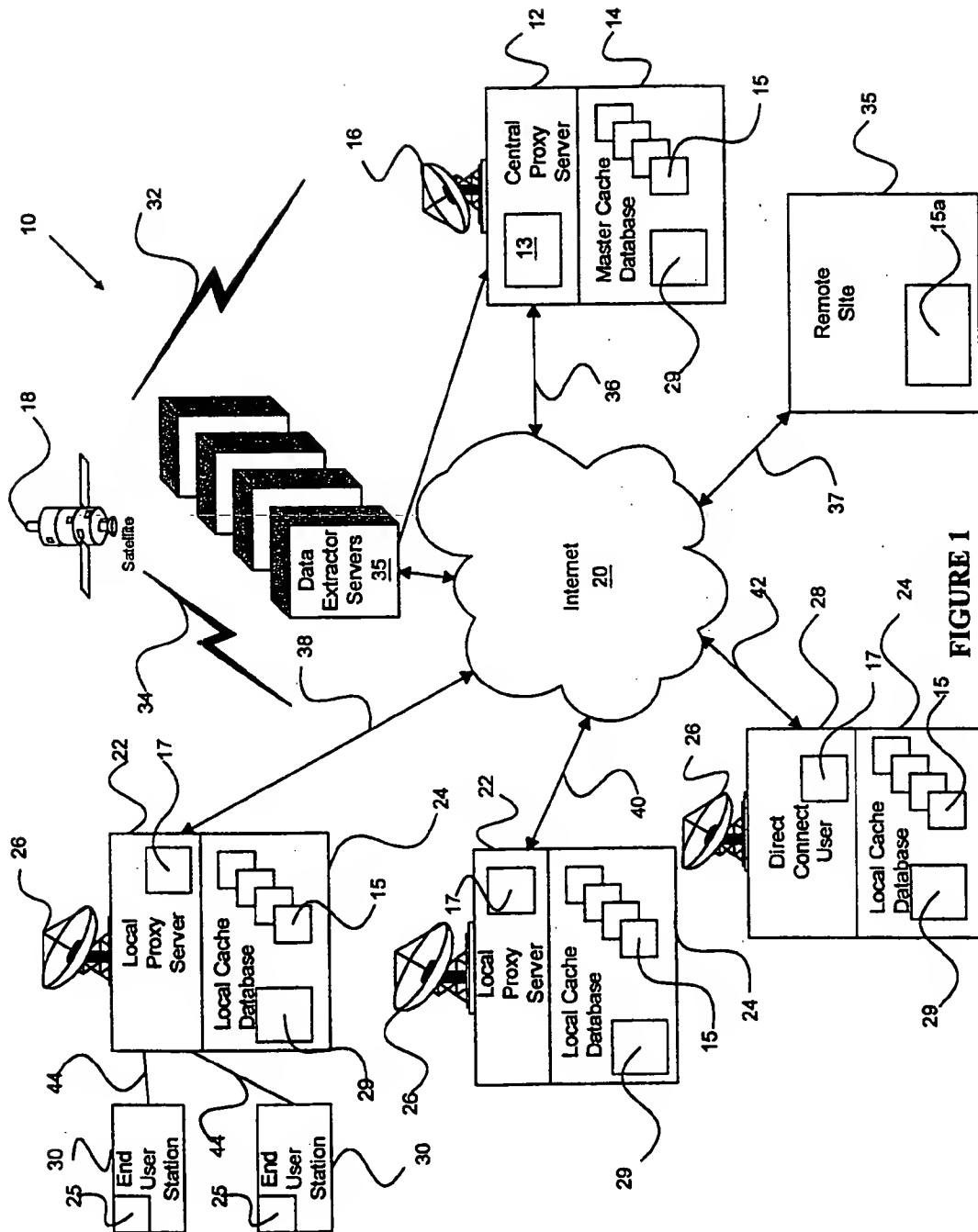


FIGURE 1

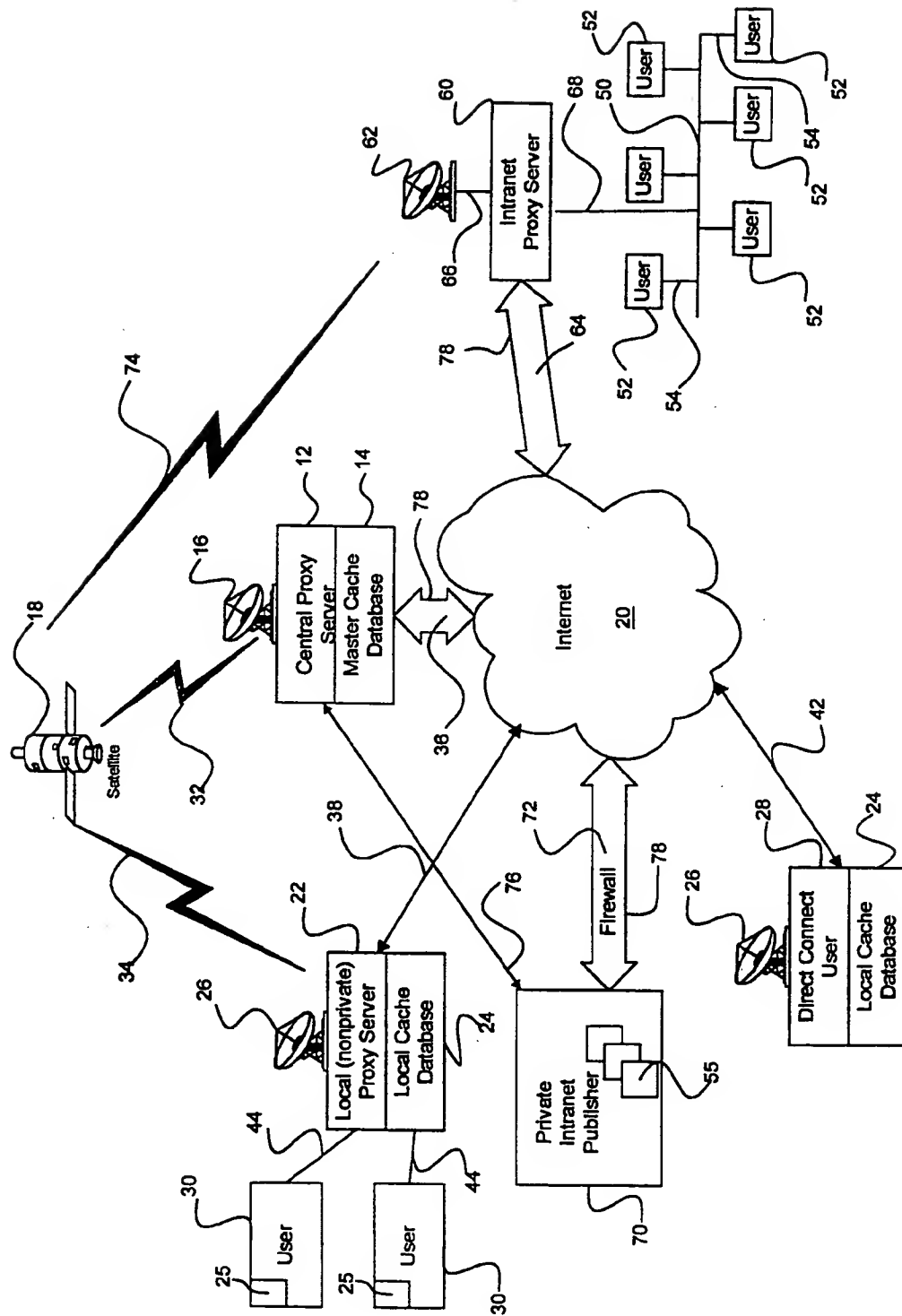


FIGURE 2

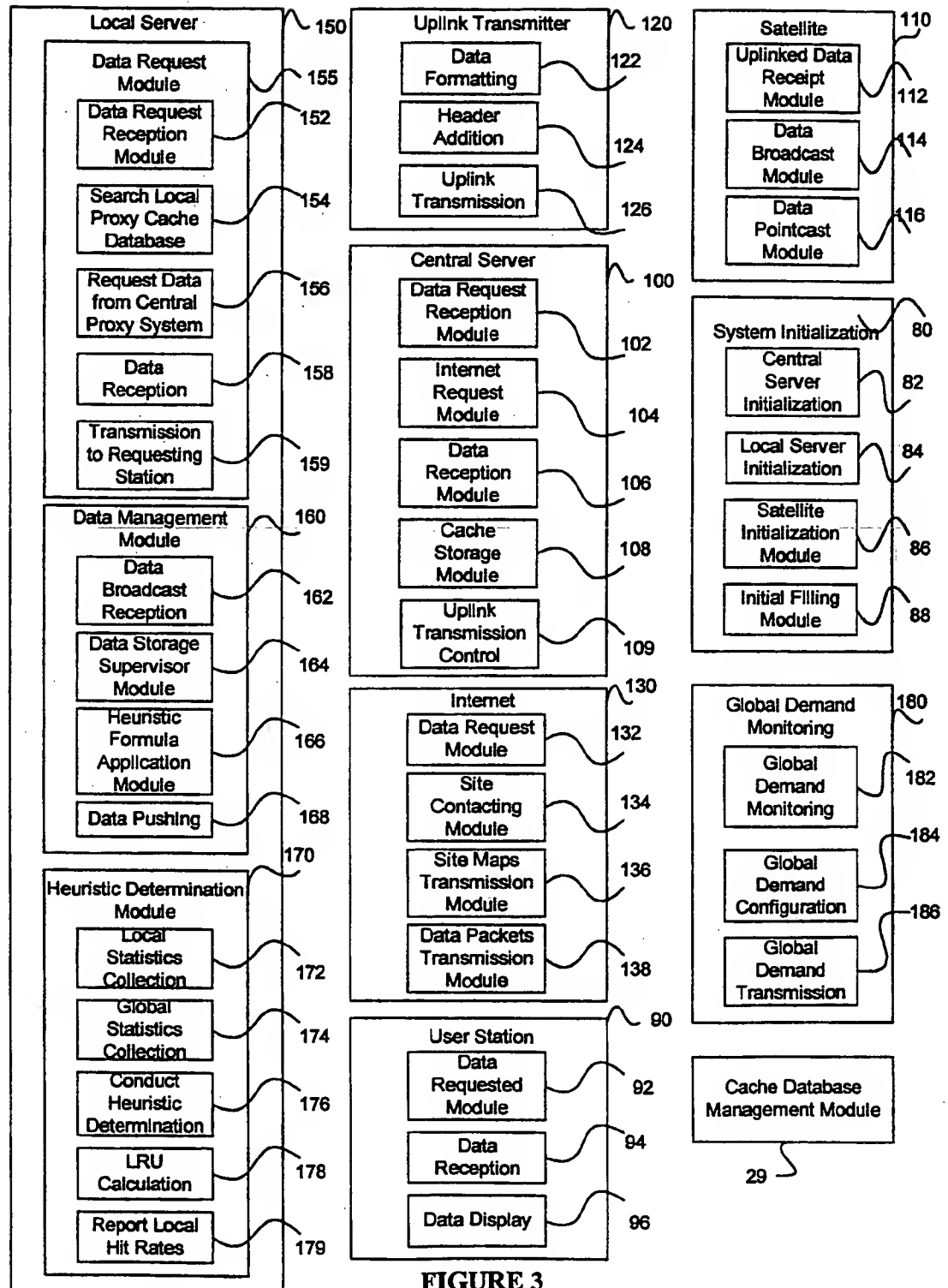


FIGURE 3

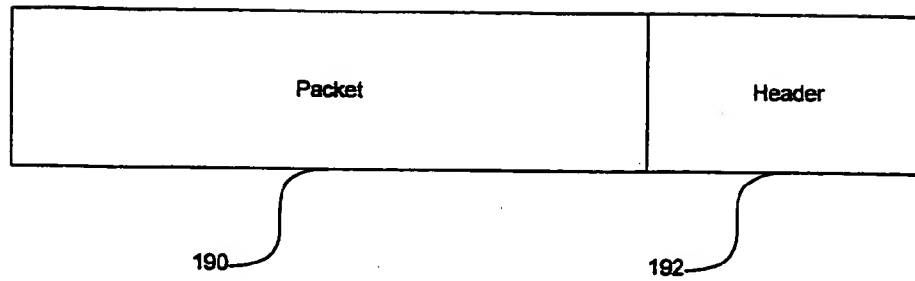


FIGURE 4

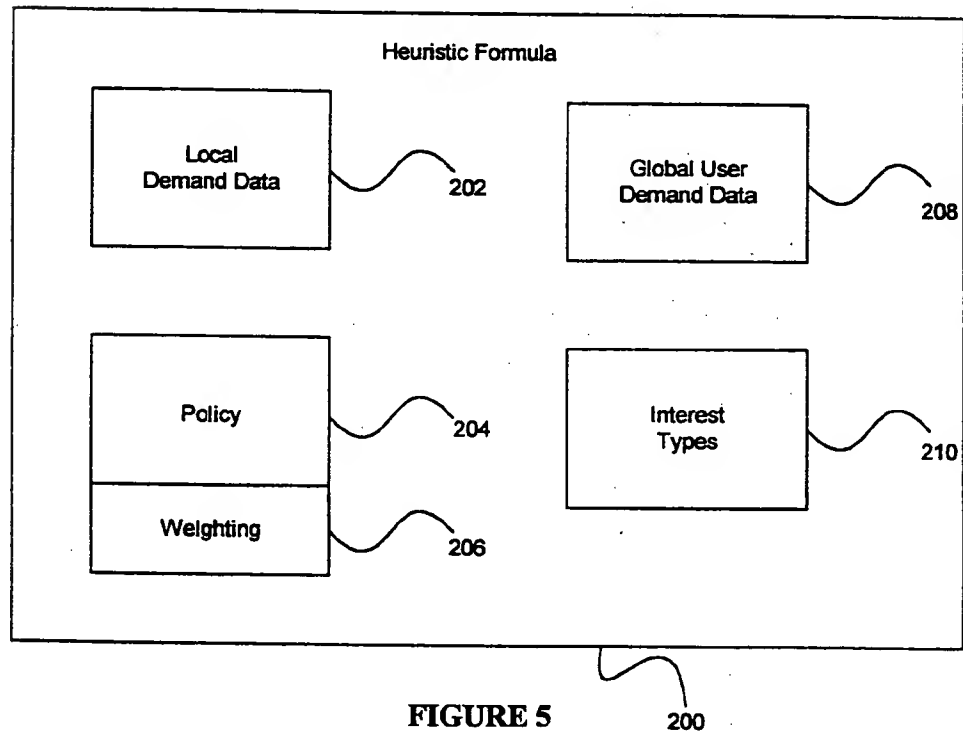


FIGURE 5

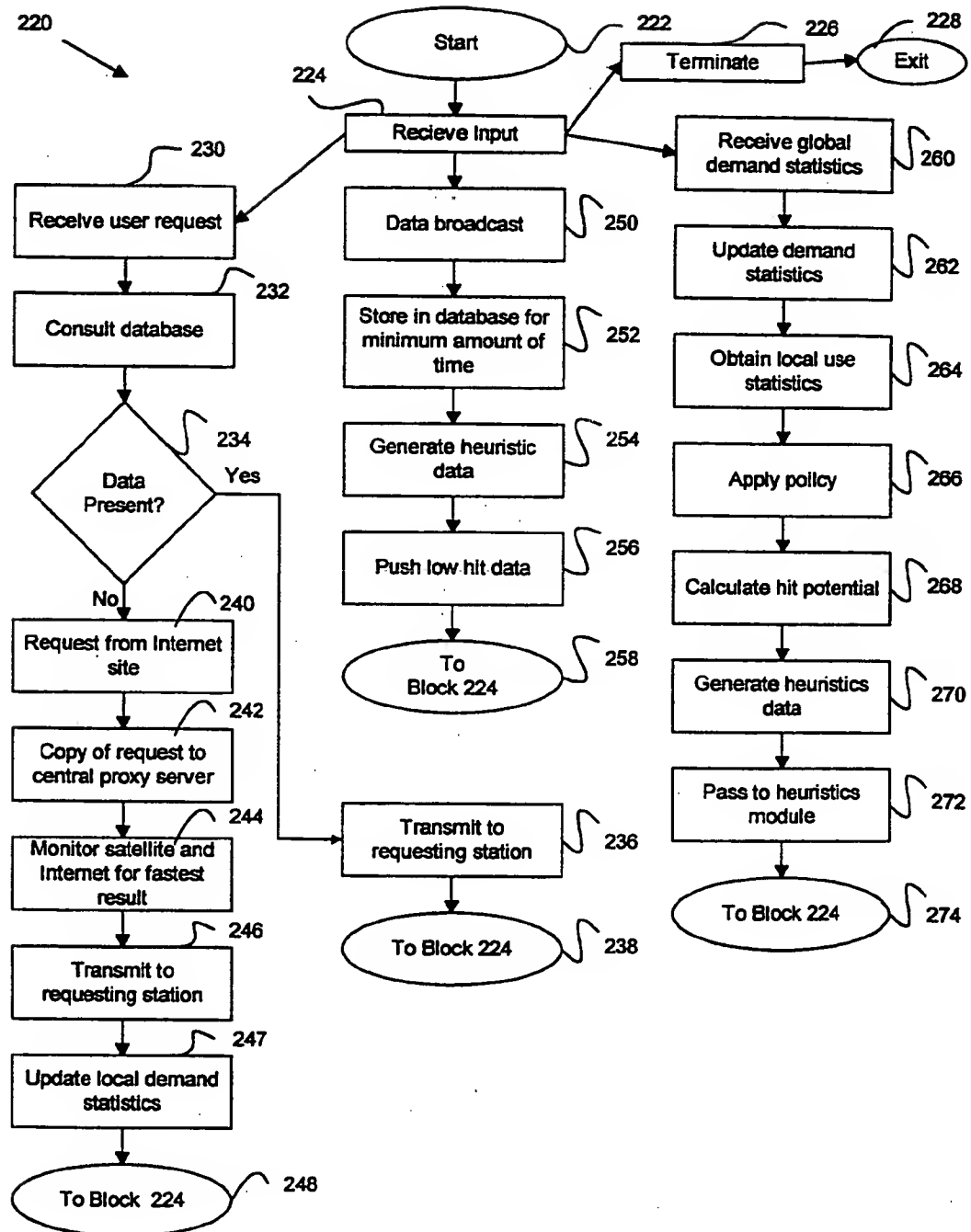


FIGURE 6

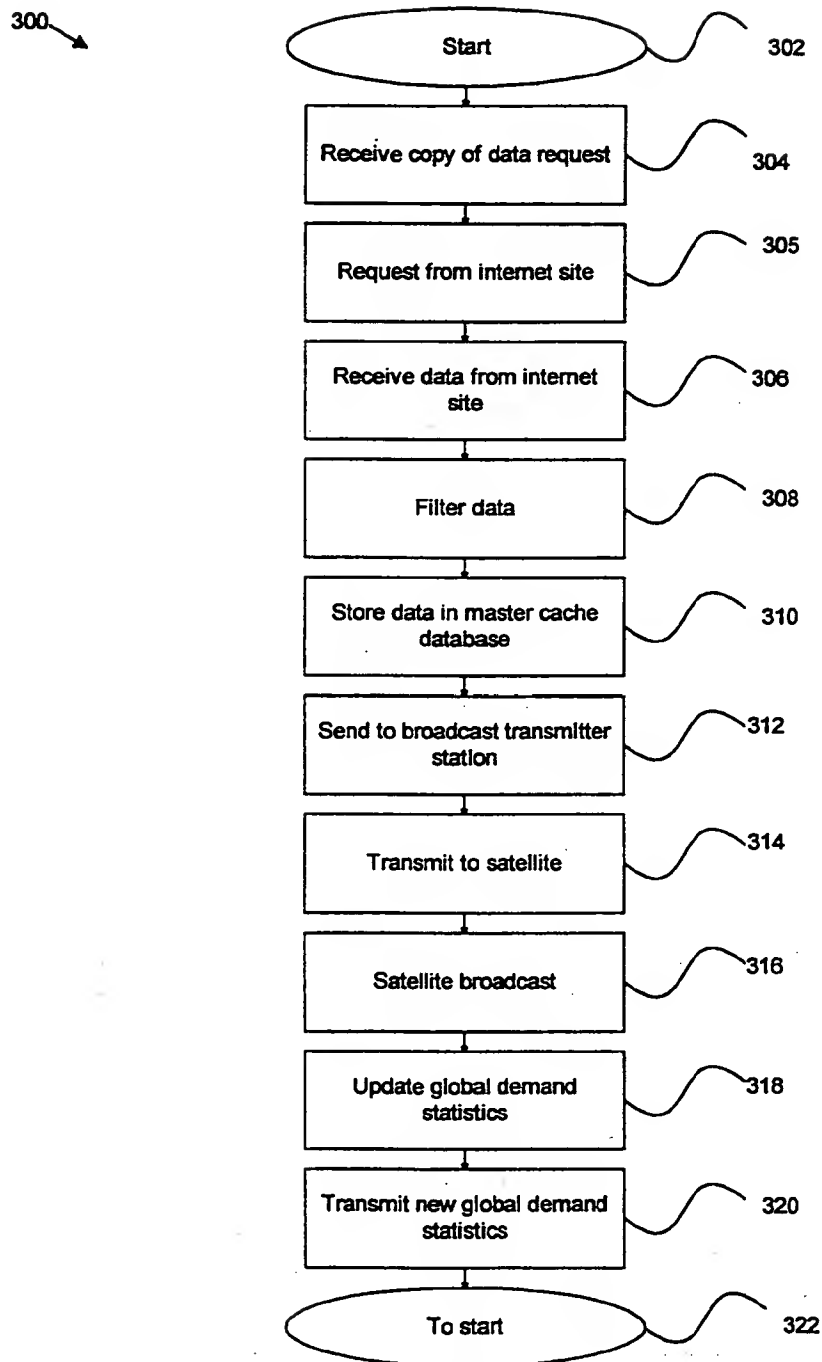
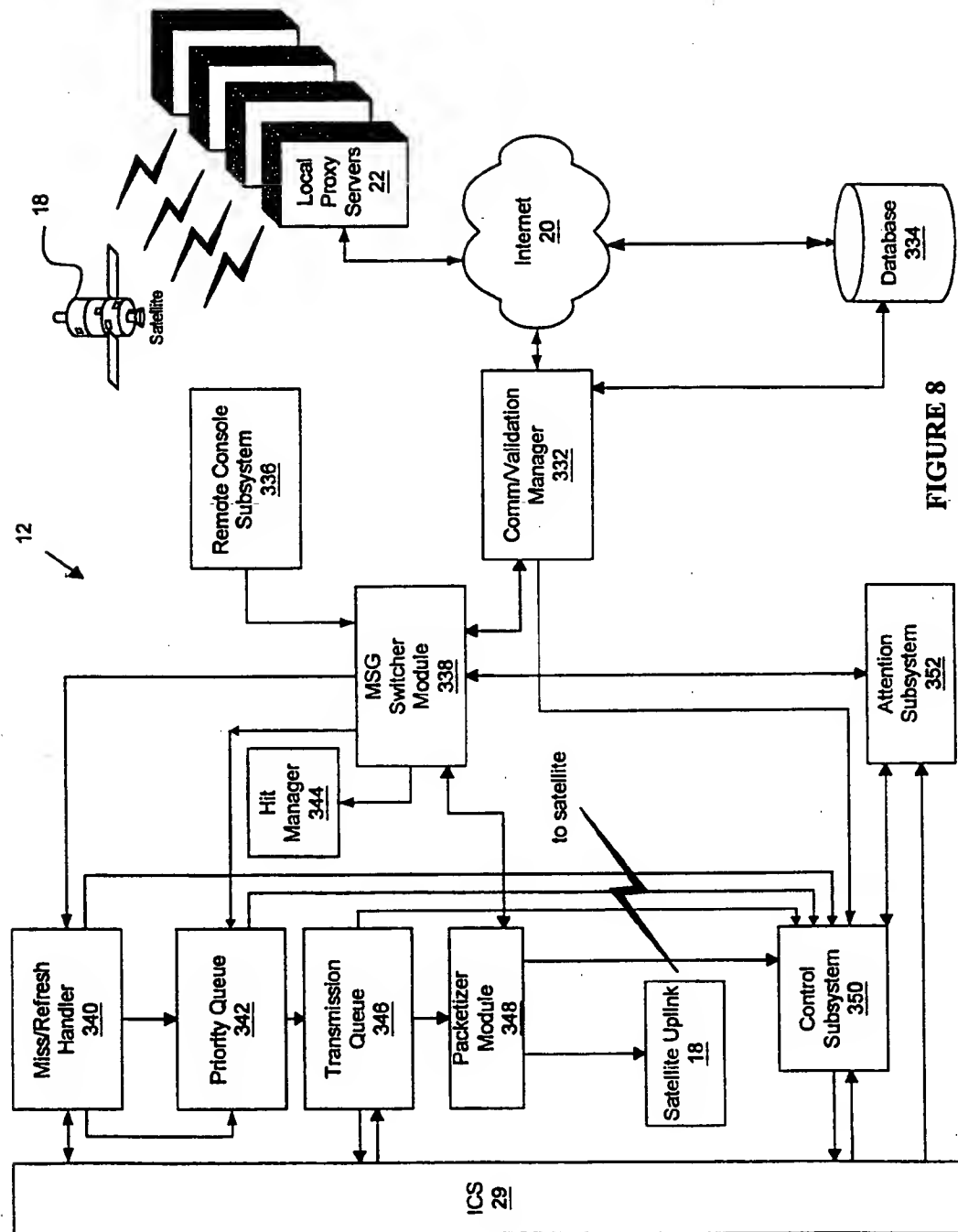


FIGURE 7



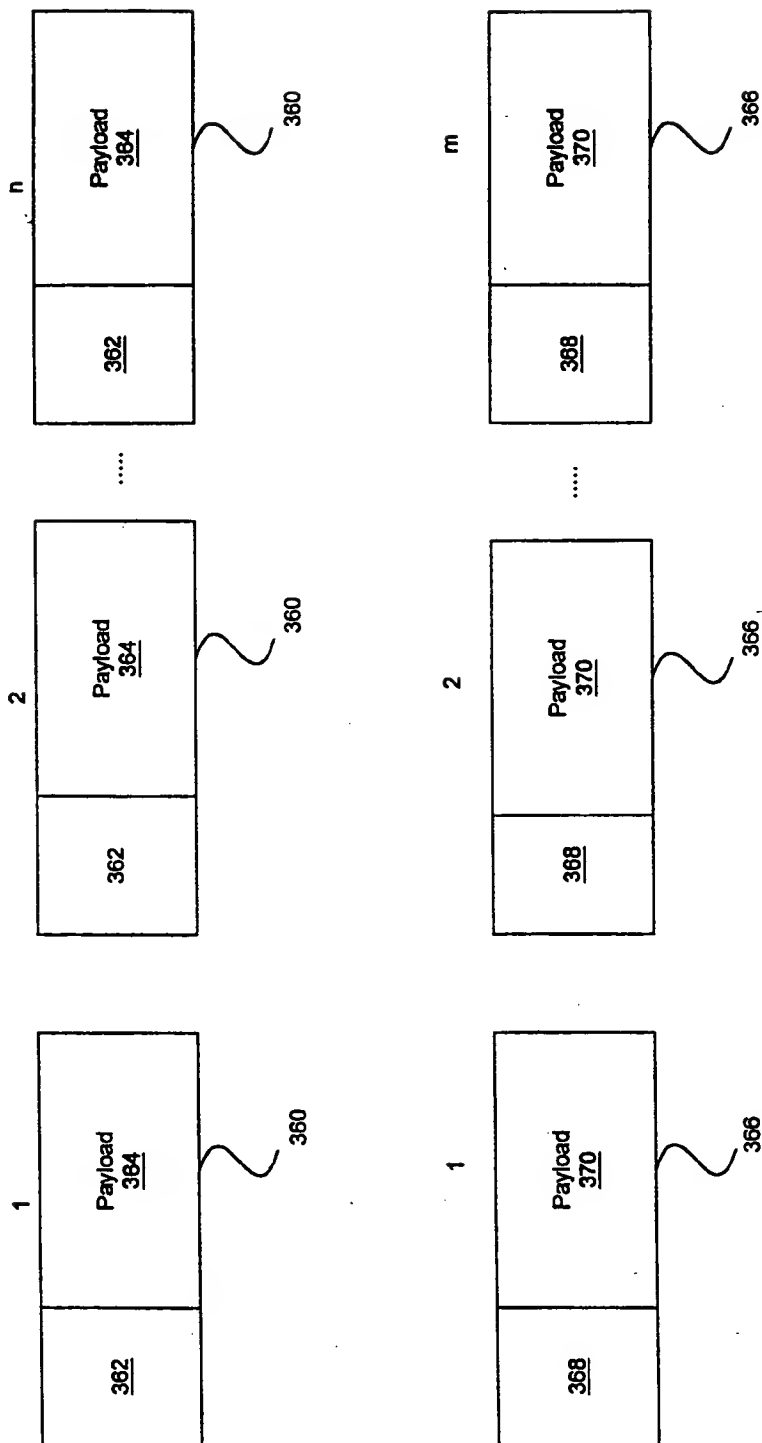


FIGURE 9

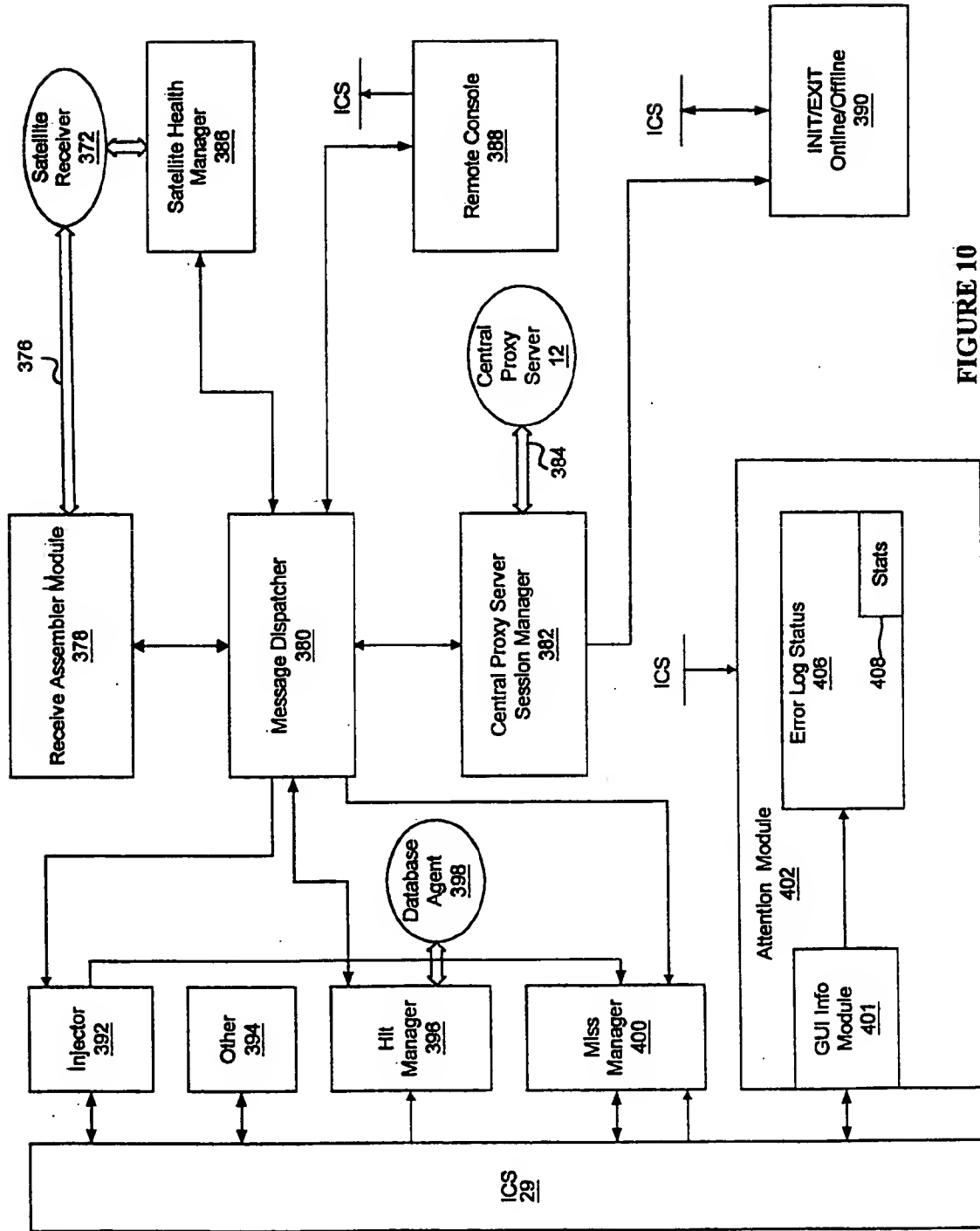
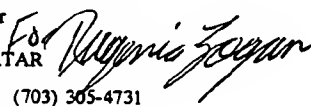


FIGURE 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/00587

A. CLASSIFICATION OF SUBJECT MATTER																				
IPC(7) : G06F 15/16 US CL : 709/217, 235																				
According to International Patent Classification (IPC) or to both national classification and IPC																				
B. FIELDS SEARCHED																				
Minimum documentation searched (classification system followed by classification symbols)																				
U.S. : 709/217, 235																				
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched																				
None																				
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)																				
WEST, EAST, IEEE																				
C. DOCUMENTS CONSIDERED TO BE RELEVANT																				
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.																		
Y,P	US 6,012,085 A (YOHE et al.) 04 January 2000, col. 5-12	1-40																		
Y,P	US 6,003,082 A (GAMPPER et al.) 14 December 1999, col. 4-12	1-40																		
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.																				
<table border="0"> <tr> <td>* Special categories of cited documents:</td> <td>* T</td> <td>later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principles or theory underlying the invention</td> </tr> <tr> <td>* A* document defining the general state of the art which is not considered to be of particular relevance</td> <td>* X*</td> <td>document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</td> </tr> <tr> <td>* E* earlier document published on or after the international filing date</td> <td>* Y*</td> <td>document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</td> </tr> <tr> <td>* L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</td> <td>* &*</td> <td>document member of the same patent family</td> </tr> <tr> <td>* O* document referring to an oral disclosure, use, exhibition or other means</td> <td></td> <td></td> </tr> <tr> <td>* P* document published prior to the international filing date but later than the priority date claimed</td> <td></td> <td></td> </tr> </table>			* Special categories of cited documents:	* T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principles or theory underlying the invention	* A* document defining the general state of the art which is not considered to be of particular relevance	* X*	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone	* E* earlier document published on or after the international filing date	* Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art	* L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	* &*	document member of the same patent family	* O* document referring to an oral disclosure, use, exhibition or other means			* P* document published prior to the international filing date but later than the priority date claimed		
* Special categories of cited documents:	* T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principles or theory underlying the invention																		
* A* document defining the general state of the art which is not considered to be of particular relevance	* X*	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone																		
* E* earlier document published on or after the international filing date	* Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art																		
* L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	* &*	document member of the same patent family																		
* O* document referring to an oral disclosure, use, exhibition or other means																				
* P* document published prior to the international filing date but later than the priority date claimed																				
Date of the actual completion of the international search		Date of mailing of the international search report																		
18 MAY 2000		12 JUN 2000																		
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231		Authorized officer																		
Facsimile No. (703) 305-3230		AHMAD MATAR 																		
		Telephone No. (703) 305-4731																		